

Laatua haastavat ilmiöt ohjelmistotuotannon päätöksenteossa

Jussi Paalanen

Pro gradu -tutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen osasto

Helsinki, 3. marraskuuta 2020

Tiedekunta/Osasto – Fakultet/Sektion – Faculty/Section		Koulutusohjelma – Studieprogram – Study Programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen maisteriohjelma	
Tekijä – Författare – Author			
Jussi Paalanen			
Työn nimi – Arbetets titel – Title			
Laatua haastavat ilmiöt ohjelmistotuotannon päätöksenteossa			
Ohjaajat – Handledare – Supervisors			
Antti-Pekka Tuovinen			
Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages	
pro gradu -tutkielma	3.11.2020	67 sivua	
Tiivistelmä – Referat – Abstract			
<p>On paljon tutkittua tietoa siitä, miten ohjelmistotuotannossa saadaan tuotettua laatua. Korkeaan laatuun tähtääminen ei kuitenkaan ole itsestäänselvyys: tutkijan kokemuksen mukaan IT:n ja liiketoiminnan rajapinnassa tehdään toistuvasti päätöksiä, joiden myötä laadun ideaaleista joustetaan. Lisäksi laatuun saattavat vaikuttaa tiedostamatta tehdyt ratkaisut ja matkan varrella kohdatut yllätykset.</p> <p>Tämän tutkimuksen tarkoituksena oli kartoittaa laatua haastavia ilmiöitä ohjelmistotuotannon päätöksenteossa ja tuoda siten uutta näkökulmaa alalla laadusta käytävään keskusteluun. Konkreettisina tavoitteina oli tunnistaa ja käsitteellistää kyseisiä ilmiöitä ja selvittää niiden esiintyvyyttä valikoiduissa ohjelmistotuotannon viitekehyksissä.</p> <p>Tutkimuksen tietoperusta muodostettiin laatua, IT:n hallinnointia ja päätöksentekoa käsittelevästä kirjallisuudesta sekä seuraavista ohjelmistotuotannon viitekehyksistä: ISO/IEC 250nn -laatustandardisarja SQuaRE, COBIT, IT4IT, TOGAF, SWEBOK, PMBOK. Tutkimuksen menetelmälliseksi lähtökohdaksi valittiin laadullinen tutkimusote. Aineisto kerättiin haastattelemalla kuutta pitkään IT-alalla eri tehtävissä työskennellyttä asiantuntijaa. Menetelmänä hyödynnettiin teemahaastattelua, ja aineisto analysoitiin sisällönanalyysin keinoin.</p> <p>Analyysin tuloksena löydettiin joukko laatua haastavia ilmiöitä, jotka käsitteellistettiin ja ryhmiteltiin kahdeksaksi kokonaisuudeksi: niukkuus, tiedon määrä ja laatu, tekninen velka, ulkoisen ympäristön muutos, tietojärjestelmän kriittisyys, tulevaisuudenkestävyys, omistajuus ja proaktiivisuus sekä kognitiiviset vinoumat. Käsitteiden muodostamisessa ja tulosten syventämisessä hyödynnettiin ilmiöihin liittyvää kirjallisuutta.</p> <p>Jälkimmäisessä vaiheessa käsitteitä peilattiin ohjelmistotuotannon viitekehyksiin. Keskeinen tulos oli, että työn arjessa kohdataan ohjelmistotuotannon laatuun vaikuttavia ilmiöitä, joita valitut viitekehykset eivät tunnista. Erityisesti tekninen velka ja kognitiiviset vinoumat ovat laajalti tunnettuja ja tutkittuja ilmiöitä, joita tarkastelun kohteeksi valituissa viitekehyksissä ei kuitenkaan käsitellä.</p> <p>Tutkimus toimii yhdenlaisena alkukartoituksena laatua haastaviin ilmiöihin ohjelmistotuotannon päätöksenteossa. Sen tuloksia voidaan hyödyntää IT:n ja liiketoiminnan yhteistyön tukena. Jatkotutkimusehdotuksena esitetään muun muassa löydettyjen ilmiöiden tarkastelemista syvemmin omina kokonaisuuksinaan.</p>			
ACM Computing Classification System (CCS):			
Social and professional topics,			
Professional topics,			
Management of computing and information systems,			
Information system economics			
Avainsanat – Nyckelord – Keywords			
ohjelmistotuotanto, laatu, päätöksenteko			
Säilytyspaikka – Förvaringställe – Where deposited			
Muita tietoja – Övriga uppgifter – Additional information			
Ohjelmistojärjestelmien erikoistumislinja			

Sisältö

1	Johdanto.....	1
1.1	Tutkimuksen tarkoitus, tavoite ja tutkimuskysymykset.....	2
1.2	Tutkimusraportin rakenne.....	3
2	Tietoperusta.....	5
2.1	Laatu ja ohjelmistotuotanto.....	5
2.2	IT:n hallinnointi ja päätöksenteko.....	8
2.3	Ohjelmistotuotannon viitekehyksiä.....	12
2.3.1	ISO/IEC 25000 -laatustandardisarja SQuaRE.....	12
2.3.2	COBIT.....	14
2.3.3	IT4IT.....	15
2.3.4	TOGAF.....	17
2.3.5	SWEBOK.....	19
2.3.6	PMBOK.....	20
3	Menetelmät.....	23
3.1	Laadullinen tutkimusote.....	23
3.2	Aineistonkeruu teemahaastatteluilla.....	25
3.2.1	Teema 1: Resurssit ja rajoitteet.....	28
3.2.2	Teema 2: Arkkitehtuurit.....	28
3.2.3	Teema 3: Elinkaaret.....	29
3.2.4	Teema 4: Toiminnalliset ja ei-toiminnalliset vaatimukset.....	30
3.2.5	Teema 5: Menetelmät ja lopputuotokset.....	30
3.2.6	Teema 6: Viestintä.....	31
3.2.7	Teema 7: Kognitiiviset vinoumat.....	32
3.3	Sisällönanalyysi aineiston analysointimenetelmänä.....	33
4	Tulokset.....	35
4.1	Laatua haastavat ilmiöt.....	35
4.1.1	Niukkuus.....	35
4.1.2	Tiedon määrä ja laatu.....	37
4.1.3	Tekninen velka.....	39
4.1.4	Ulkoisen ympäristön muutos.....	42
4.1.5	Tietojärjestelmän kriittisyys.....	43
4.1.6	Tulevaisuudenkestävyys.....	45
4.1.7	Omistajuus ja proaktiivisuus.....	47
4.1.8	Kognitiiviset vinoumat.....	49
4.2	Ilmiöiden esiintyminen ohjelmistotuotannon viitekehysissä.....	50

5	Pohdinta.....	53
5.1	Laatua haastavien ilmiöiden tarkastelua	53
5.2	Tutkimuksen arviointi.....	56
5.3	Jatkotutkimusehdotukset.....	58
	Lähteet.....	60

1 Johdanto

Tutkimus on tuottanut paljon tietoa siitä, miten ohjelmistotuotannossa voidaan tehdä hallitusti laadukasta tulosta. Ylätasolla puhutaan esimerkiksi laatukulttuurista, laatujohtamisesta, laatujärjestelmistä [LaA18] tai kokonaisvaltaisen laadunhallinnan ajattelumallin soveltamisesta informaatioteknologian (IT) toimialalle [KTS14]. Laadun puolestapuhujat pyrkivät rakentamaan eetosta, joka kannustaa päätöksentekotilanteissa investoimaan laatuun, perusteilla kuten että hyvä laatu maksaa itsensä takaisin [JuG98, Ste95].

Laatutekniikat käsittävät laatu työn kokonaisvaltaisena ammattialana, jossa suunnitellaan, varmistetaan, testataan ja kehitetään laatua [Tia05]. Laadun arvioimiseksi ja haluttuun suuntaan kehittämiseksi on tarjolla laatumalleja, joilla mallinnetaan tarkasteltavien kohteiden kuten ohjelmistojen, arkkitehtuurien, datan, tiedon, dokumenttien ja prosessien laatuominaisuuksia, ja samalla otetaan filosofistakin kantaa siihen monitahoiseen asiaan, mitä laatu oikeastaan on ja voiko sitä määritellä vai ei [Kan03, AnJ16]. Laadun tuottamiseen ja varmistamiseen ohjelmistotuotannossa on tarjolla monenlaisia menetelmiä ja työkaluja. On kääntäjiä, ohjelmistojen käännösaikaisia tarkastajia, koodiauditointeja, retrospektiivejä, katselmointeja ja versionhallinnan *pull request* -läpikäyntejä. On testausta, testauksen automatisointia, käyttäytymislähtöistä kehitystä ja hyväksymistestauslähtöistä kehitystä.

Menetelmäkirjon ja ohjelmistotuotannon laatua luotaavan tutkimustiedon pohjalta voidaan olettaa, että korkean laadun saavuttamiseksi on olemassa hyvät mahdollisuudet. Miten mahdollisuuksiin tartutaan ja miten ne todentuvat käytännössä, on toinen asia: tie korkeaan laatuun – sikäli kuin sille ylipäättään päätetään lähteä – voi olla mutkainen, eikä tietämys aiheesta sen enempää kuin tekninen osaaminenkaan vielä takaa laatustandardien ja -mittareiden valossa timanttista lopputulemaa. Ohjelmistojen laatuun ja laadun kehittämiseen perehtynyt tutkija ja kirjailija Dick Fairley on todennut: "Quality is free but [the real question] is [if] you're willing to invest in it" [Hin07]. Laatu ei synny itsestään – siihen täytyy panostaa.

Tutkijan omassa työssä IT-konsulttina ja audittoijana laatu näkökulma on jatkuvasti läsnä. Reilun 20 vuoden aikana hankitut omakohtaiset kokemukset IT-alalta ovat osoittaneet, että IT:n ja liiketoiminnan rajapinnassa kohdataan ilmiöitä, jotka tavalla tai toisella haastavat laatua. Ne voivat näyttäytyä tietoisina päätöksinä tinkiä laadusta, mutta yhtä hyvin laatuun saattavat vaikuttaa tiedostamatta tehdyt ratkaisut ja matkan varrella kohdatut yllätykset. Kysymys siitä, milloin ja miksi laadun ideoista joustetaan, toimii alkusysäyksenä tämän tutkimuksen tekemiselle.

1.1 Tutkimuksen tarkoitus, tavoite ja tutkimuskysymykset

Tutkimuksen aihevalinta kumpuaa tutkijan oman työn todellisuudesta, mutta sille on myös laajemmat perusteet, jotka on johdettavissa IT:n ja liiketoiminnan yhteensovittamisesta. Organisaation päätöksentekotilanteissa IT:n ja liiketoiminnan näkökulmat ja tarpeet kohtaavat. Teknologiasta vastuussa olevat johtajat ja arkkitehdit lähestyvät asioita laadullisten tarpeiden kuten tietoturvan, suorituskyvyn ja teknologiavalintojen kustannustehokkuuden näkökulmasta. Liiketoimintanäkökulmaa edustavien ihmisten huolena on puolestaan yrityksen tuottavuus, kasvu, markkinaosuus ja strategia, monesti kvartaalitalouden rytmittämänä. Haasteena on pyrkiä tekemään laadukkaita päätöksiä, jotka täyttävät molempien näkökulmien tarpeet optimaalisella tavalla. Käytännössä joudutaan kuitenkin tekemään kompromisseja, joissa kaikkia IT:n laatutarpeita ei saada täytettyä [HäL07].

Tutkimuksen tarkoituksena on kartoittaa laatua haastavia ilmiöitä ohjelmistotuotannon päätöksenteossa ja osallistua siten ymmärryksen lisäämiseen puolin ja toisin IT:n ja liiketoiminnan välisessä kanssakäymisessä. Tutkimus rajataan arkkitehtuuritason päätöksiin erotuksena esimerkiksi sovelluskehityksen yhteydessä tehtäviin suunnittelutason (*design*) päätöksiin. Kiinnostus suuntautuu siten ylätasoon ja – tutkijan kokemuksen mukaan –sosioteknisesti monimutkaisempiin ilmiöihin kuin vaikkapa ohjelmointi- ja määrittelyvirheisiin tai suunnittelutason parantelumalleihin (*anti-patterns*). Arkkitehtuuritasoista mukaan kelpaavat kaikki, aina yksittäisistä ohjelmistojen sovellusarkkitehtuureista järjestelmäintegraatioarkkitehtuureihin, tietoarkkitehtuureihin ja organisaatioiden kokonaisarkkitehtuureihin. Tutkimuksen pääasiallisena organisaatiokontekstina ovat monitoimijaympäristöt, joiden keskiössä ovat sellaiset yritykset ja julkishallinnon organisaatiot, jotka hankkivat paljon IT:tä ja IT-palveluita tukemaan liiketoimintaprosessejaan, mutta joiden ansaintalogiikka ei ole olla ohjelmistotuotantotalo.

Koska tutkimus on luonteeltaan kartoittava, uusia näkökulmia ja ilmiöitä etsivä, hyödynnetään sen tekemisessä laadullista tutkimusotetta. Aineistonkeruumenetelmänä käytetään teemahaastattelua, ja haastatteluaineisto muodostaa tutkimuksen varsinaisen perustan ja lähtökohdan. Käytännössä kuudelta Suomessa IT-alalla pitkään työskennelleeltä asiantuntijalta kerätään kokemusperäistä tietoa laatua haastavista ilmiöistä. Koska valittua aihetta – laatua haastavia ilmiöitä, laadun kilpailijoita tai vastavoimia ohjelmistotuotannon päätöksenteossa

– suoraan käsittelevää teoriaa ei ollut löydettävissä, tietoperustassa keskitytään määrittelemään keskeiset käsitteet ja esittelemään valikoituja ohjelmistotuotannon laatua, päätöksentekoa ja hallintaa tarkastelevia viitekehyksiä.

Tutkimuksen konkreettisena tavoitteena on käsitteellistää haastatteluissa esiin nousevia ilmiöitä ja tuoda näin uutta näkökulmaa laadusta käytävään keskusteluun. Käsitteitä voidaan hyödyntää esimerkiksi tunnistamaan tilanteita, joissa saatetaan olla tekemässä laadun kannalta arveluttavia päätöksiä. Lisäksi tavoitteena on selvittää, missä määrin ilmiöitä kuvaavia käsitteitä huomioidaan valikoiduissa ohjelmistotuotannon viitekehyksissä. Tutkijan henkilökohtaisena tavoitteena on paitsi hankkia laajempi ymmärrys tutkimukseen valikoiduista viitekehyksistä, niiden sisällöstä ja tarkoituksesta, myös selkiyttää oman työn tarpeisiin, mitkä tekijät voivat vaikuttaa siihen, ettei ohjelmistotuotannossa valita laatua. Tutkimuksen myötä tutkija toivoo saavansa oman ja kollegojensa työn tueksi uusia ideoita laadun perustelemiseen, tuottamiseen ja varmistamiseen.

Tutkimuksessa haetaan vastauksia seuraaviin tutkimuskysymyksiin:

- TK1: Mitä laatua haastavia ilmiöitä ohjelmistotuotannon päätöksenteossa on tunnistettavissa haastatteluaineiston perusteella?
- TK2: Tunnistavatko tutkimukseen valitut ohjelmistotuotannon viitekehykset kyseisiä ilmiöitä?

1.2 Tutkimusraportin rakenne

Tutkimuksessa hyödynnetty teoreettinen aines jakautuu kahteen osaan, joista ensimmäisen muodostaa luvun kaksi tietoperusta. Se on tiivis johdanto tutkimuksen empiriseen osaan. Tarkastelun kohteena ovat tutkimuksen kannalta keskeiset käsitteet, joista ensimmäiseksi käsitellään laatua ohjelmistotuotannon kontekstissa. Tämän jälkeen paneudutaan IT:n hallintointiin ja päätöksentekoon. Luvun lopuksi esitellään joukko ohjelmistotuotannon laatuun, päätöksentekoon, johtamiseen ja hallintaan linkittyviä alalla tunnettuja viitekehyksiä ja käsikirjoja: ISO/IEC 25000 -laatustandardisarja SQuaRE, COBIT, IT4IT, TOGAF, SWEBOK, PMBOK.

Kolmas luku käsittelee tutkimuksessa tehtyjä menetelmällisiä valintoja. Laadullinen tutkimusote tuodaan esille tutkimuksen tekemistä ohjaavana laajana näkökulmana. Tämän jälkeen tarkastellaan teemahaastattelua aineistonkeruumenetelmänä ja kuvataan tässä tutki-

muksessa käytetyt teemat ja niiden yhteydessä hyödynnetyt, keskustelun herättäjinä toimineet kuvat. Kolmannen luvun viimeisessä osassa kerrotaan haastatteluaineiston analysoimisesta sisällönanalyysin keinoin.

Luvussa neljä esitellään työn tulokset ja vastataan tutkimuskysymyksiin. Tutkija kuvaa, mitä laatua haastavia ilmiöitä haastatteluaineistosta on tunnistettavissa ja minkälaisilla käsitteillä ilmiöitä nimitetään alan kirjallisuudessa. Tuloksia syvennetään ja uutta kirjallisuutta haetaan tukemaan haastatteluissa löydettyjä ilmiöitä. Tämä muodostaa tutkimuksessa hyödynnetyn teorian toisen osan. Sitä tarvitaan, sillä kuten edellä on tuotu esille, aihepiiriin ei tiettävästi liity suoraan teorioita, joita voitaisiin testata empiirisellä aineistolla. Sen sijaan neljännen luvun lopuksi tarkastellaan, miten ilmiöitä kuvaavat käsitteet ovat edustettuina tietoperustaan valituissa kuudessa ohjelmistotuotannon viitekehyksessä.

Pohdinta on työn viides ja viimeinen luku, jossa tehdään yhteenvetoa tutkimuksen annista. Luvussa tutkija esittelee tutkimuksen aikana saamansa keskeiset oivallukset. Arvioinnin yhteydessä pohditaan tutkimuksen luotettavuutta, aineiston riittävyyttä ja valittuja menetelmiä, tutkimukseen liittyviä rajoituksia ja haasteita sekä asetettujen tavoitteiden saavuttamista. Työn päättävät tutkijan ehdotukset jatkotutkimusaiheiksi.

2 Tietoperusta

Tietoperustan tarkoituksena on ensinnäkin muodostaa esiymmärrys tutkimukseen liittyvistä keskeisistä käsitteistä, joita ovat laatu, ohjelmistotuotanto ja päätöksenteko. Tietoperustan jälkimmäinen osa koostuu joukosta ohjelmistotuotannon viitekehyksiä, joita organisaatiot voivat käyttää laadun määrittämisen ja mittaamisen sekä päätöksentekoprosessien tukena.

2.1 Laatu ja ohjelmistotuotanto

Laatu on monitahoinen käsite, jolle on tarjolla runsaasti erilaisia määritelmiä. Niitä voidaan ryhmitellä esimerkiksi Anttilan ja Jussilan mukaan seuraavasti: Tuoteperustaisissa määritelmässä laatu määritetään tuotteen mitattavien ominaisuuksien kautta. Tuotantoperustaisissa määritelmässä laatu on vaatimusten täyttämistä ja täyttymistä. Rahallisissa arvoperustaisissa määritelmässä laatu on käyttöarvon ja hinnan suhde. Reaalitaloudellisissa arvoperusteisissa määritelmässä laatu on tuotteen hyötyarvo eli käyttäjän subjektiivinen kokemus saamastaan hyödystä tuotteen eliniän aikana. Heurististen ja myyttisten määritelmien mukaan laatua ei voida mitata eikä edes määritellä suljetusti [AnJ16].

Lexico-sanakirja antaa laadulle seuraavan määritelmän: "The standard of something as measured against other things of a similar kind; the degree of excellence of something." [Lex20]. ISO 9000 -standardin viimeisimmän version mukaan laatu on sitä, missä määrin kohteen luontaiset ominaisuudet täyttävät kohteelle asetetut vaatimukset [ISO90]. Crosbylla on lainattu paljon ajatusta "quality is conformance to requirements". Sen mukaan laatu syntyy huolellisesti laadittujen vaatimusten noudattamisesta. Näin ollen laatu on oikeastaan ilmaista, koska se sisältyy valmistuskustannuksiin [Ste95]. Juranin laatukäsikirjassa laatu hahmotetaan puolestaan yhdistelmäksi kahta erilaista laatua: tuotteen ominaisuuksia, jotka vastaavat asiakkaiden tarpeita, sekä tuotteen "vapautta heikkouksista" (*freedom from deficiencies*). Tuotteen ominaisuuksien kehittäminen yleensä nostaa kustannuksia, ja tuotteen heikkouksien karsiminen yleensä vähentää kustannuksia pitkällä aikavälillä [JuG98].

Tunnettu on niin ikään Garvinin viiden näkökulman määritelmä tuotelaadusta: Asiakaskeskeinen näkökulma tarkastelee käyttäjien tarpeiden ja mieltymysten täyttämistä. Valmistuskeskeinen näkökulma kiinnittää huomion siihen, miten hyvin tuote noudattaa ennalta määritettyjä vaatimuksia ja suunnitelmia. Tuotekeskeisessä näkökulmassa laatu määrittyy tuotteen ominaisuuksien kautta. Arvokeskeinen näkökulma tarkastelee laatua rahallisen arvon

avulla eli on kiinnostunut siitä, mitä tuotteesta ollaan valmiita maksamaan. Lopuksi transkendenttisen näkökulman valossa laatu näyttyy absoluuttisena ja yleisesti tunnistettavana erinomaisuutena, jota ei pystytä tarkkaan määrittelemään [Gar84].

Kansainvälinen tekniikan alan järjestö Institute of Electrical and Electronics Engineers (IEEE) määrittelee käsitteen ohjelmistotuotanto (*software engineering*) tarkoittavan järjestelmällistä, kurinalaista ja mitattavissa olevaa toimintaa, jolla kehitetään, operoidaan ja ylläpidetään ohjelmistoja. Sen voidaan ajatella pitävän sisällään suuren määrän erilaisia kokonaisuuksia, kuten vaatimusten määrittelyä, suunnittelua, ohjelmistojen toteutusta, testausta ja ylläpitoa [IEE90].

Ohjelmistotuotannon alalla on käytössä omia toimialakohtaisia määritelmiä laadulle. ISO/IEC:n ohjelmistotuotelaadun standardi 25010 määrittelee ohjelmistotuotteen laadun olevan sitä, missä määrin ohjelmisto täyttää määritellyt ja määrittelemättä jätetyt tarpeet, kun ohjelmistoa käytetään tietyissä olosuhteissa [ISO11a]. ISO/IEC:n datan laatustandardi 25012 antaa vastaavan määritelmän datalle [ISO11b]. Samaan tapaan mitä tahansa muutakin ohjelmistotuotantoalan kohdetta kuten dokumenttia, sovellusarkkitehtuuria tai tuotantoprosessia voidaan tarkastella laatu näkökulmasta, kunhan ensin sovitaan, miten juuri kyseisen kohteen laatu määritellään. Määrittelyssä voidaan törmätä erilaisiin haasteisiin. Jos esimerkiksi määritellään laadun olevan sitä, kuinka hyvin eksplikoitavat vaatimukset täytetään, pitää jotenkin erikseen huomioida se, että vaatimuksissa voi olla virheitä.

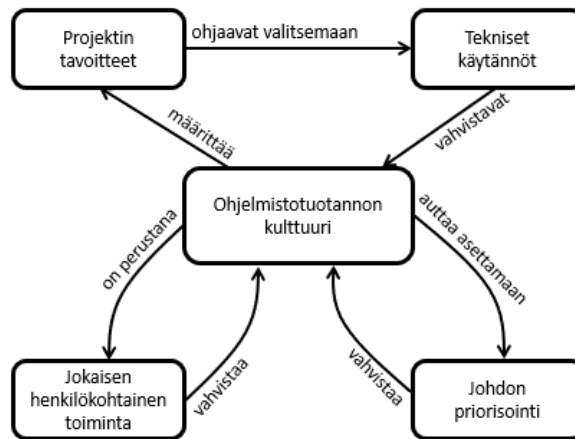
Ohjelmistotuotannon prosesseilla on vaikutusta tuotosten laatuun [HaC01]. Tämä toimii motivaationa mitata ja kehittää myös prosessien laatua. Software Process Improvement (SPI) on ohjelmistoprosessien kehitysmetodologia, jota noudattavia viitekehyksiä ovat muun muassa kansainvälinen standardiperhe ISO 33xxx - Software Process Improvement and Capability Determination (SPICE), siitä pienempiä organisaatioita tai projektitiimejä varten laadittu variantti ISO/IEC 29110 Systems Engineering International Standard for Very Small Entities sekä Software Engineering Instituten (SEI) Capability Maturity Model Integration (CMMI). SPI-metodologia pyrkii esittelemään tehtäviä, työvälineitä ja tekniikoita, joiden avulla on mahdollista suunnitella ja tehdä kehittämistyötä, joka tähtää esimerkiksi tehostamiseen, korkeampaan laatuun tai kustannusten vähentämiseen [IGI20]. SPI-mallien käyttöönotto on kuitenkin aikaavievää ja kallista, ja SPI-käyttöönottohankkeet epäonnistuvat usein [Nia06].

Jos halutaan määrittää laadun taso ja asettaa laadulle tavoitteita, laatu on kyettävä määrittelemään mitattavissa olevien ominaisuuksien kautta [FeB14, s. 441]. Toisin sanoen tarvitaan tarkka mittausprotokollan kuvaus ja työkalusto, jolla kunkin ominaisuuden mittaaminen tehdään. Erilaisten laatumallien yksi tavoite on pyrkiä vastaamaan toimialan tarpeeseen saada vertailukelpoisia mittaustuloksia ohjelmistotuotannon laadusta [FeB14]. Mittaaminen ei kuitenkaan ole ongelmatonta. Malleja on kritisoitu siitä, että ne jättävät määrittelemättä yksiselitteisesti, miten kunkin laatuattribuutin tai ominaisuuden mittaaminen tulee tehdä. Lisäksi on todettu tapauksia, joissa eri työkaluilla tehdyt mittaukset ovat tuottaneet eri tuloksia samalla lähdeaineistolla, vaikka mittausprosessit on määriteltä identtisiksi [LLL08].

Jones ja Bonsignour toteavat, että ohjelmistotuotannon toimialalla laadun mittauskäytännöt ovat yleisesti heikolla tolalla. Tämä on heidän mukaansa hämärtänyt laadun ja talouden välistä yhteyttä ja saanut aikaan, että monet johtajat ja jopa laatuasiantuntijat ovat taipuvaisia pitämään laatua yksinkertaisesti kuluna: työtä hidastavana ja kustannuksia kasvattavana tekijänä. Käytännön projekteja analysoimalla on kuitenkin pystytty keräämään todisteita siitä, että korkea laatu on päinvastoin yhteydessä keskimääräistä pienempiin kustannuksiin ja lyhyempikestoisiin projekteihin [JoB12].

Fenton ja Neil pohtivat syitä siihen, miksi ohjelmistometriikan käyttö ohjelmistotuotannossa on edelleen vähäistä, vaikka ohjelmistometriikka tutkimusalanä on ollut tunnettu jo 1960–70 luvuilta lähtien. Heidän mukaansa ohjelmistometriikoiden käyttöä ei ole riittävästi kohdennettu tuottamaan käyttökelpoista kvantitatiivista tietoa päätöksenteosta ja johtamisesta vastaaville sidosryhmille. He suosittelevat, että johtamisen ja päätöksenteon tueksi aletaan tarjota tietoa, joka tuo esille ilmiöiden välisiä syy–seuraussuhteita ja ilmiöihin liittyvää epävarmuutta ja joka perustuu useisiin, myös subjektiivisiin tietolähteisiin [FeN00].

Kulttuurin merkitys nousee esille ohjelmistotuotannon laadun yhteydessä: kulttuuri vaikuttaa laatuun [ApL05]. Wiegers kuvaa (kuva 1, s. 8), kuinka tekniset käytännöt, jokaisen henkilökohtainen toiminta ja se miten johtamisessa priorisoidaan asioita, vahvistavat aina kulttuuria johonkin suuntaan. Kulttuuri puolestaan toimii yhtenä ohjaavana tekijänä edellä mainituille. Näin kulttuurin ja toiminnan välille syntyy vuorovaikutuksen silmukoita [Wie96]. Kulttuurilla on merkittävä painoarvo organisaation prosessien kehityksessä: organisaatio ei voi rahalla ostaa prosesseja, jotka tuovat mukanaan laadun [ApL05]. Johto voi asioiden priorisointien kautta kannustaa laadun tekemiseen ja antaa siihen riittävät resurssit. Jos tätä ei tapahdu, jokaisella on mahdollisuus henkilökohtaisen toimintansa kautta auttaa johtoa näkemään laadun arvon.



Kuva 1. Ohjelmistotuotannon kulttuuri [mukaillen Wie96]

2.2 IT:n hallinnointi ja päätöksenteko

Tutkimuksen kannalta on ymmärrettävä päätöksentekoa riittävästi, jotta voidaan käsitellä sitä tutkimuskysymyksiin vastaamisen vaatimalla tasolla. Jotta päätöksentekoa voidaan ymmärtää, on lähestyttävä sitä laajemmasta kontekstista, hallinnoinnin ja johtamisen kautta, ja selvitettävä niihin olennaisesti liittyvät tulosvastuullisuuden ja vastuunjaon käsitteet.

Weill ja Ross määrittelevät IT:n hallinnoinnin (*governance*) prosessiksi, jolla luodaan viitekehys IT:tä koskevalle päätöksenteolle ja vastuuttamiselle organisaatiossa. Sen pyrkimyksenä on ohjata IT:n käyttöä suuntaan, joka tukee organisaation strategiaa. IT:n johtaminen (*management*) on puolestaan hallinnoinnin mahdollistama prosessi, jossa varsinaiset IT:tä koskevat päätökset tehdään ja jossa huolehditaan päätösten toteuttamisesta käytännössä [WeR04]. Hallinnointi näyttäytyy siten suuntaa antavana ja arvioivana ylätason kokonaisuutena, kun taas johtaminen ottaa kantaa operatiivisen tason toimintaan.

IT Governance Institutin mukaan IT:n hallinnointi on pohjimmiltaan kiinnostunut kahdesta asiasta: IT:n arvontuotosta liiketoiminnalle ja riskien vähentämisestä. Ensimmäiseen tarvitaan IT:n ja liiketoiminnan yhteensovittamista ja jälkimmäiseen tulosvastuullisuuden mekaniisin sulauttamista organisaatioon. Molemmat edellyttävät riittäviä resursseja sekä tulosten mittaamista ja valvontaa [Boa03]. De Haes ja Van Grembergen käyttävät IT:n hallinnoinnista nimeä Enterprise Governance of IT (EGIT) ja IT:n johtamisesta nimeä Business/IT Alignment (BITA) pyrkimyksenään vahvistaa liiketoiminnan roolia IT:n hallinnoinnissa ja johtamisessa (kuva 2, s. 9) [HaG09].



Kuva 2. Enterprise Governance of IT (EGIT) [mukaiillen HaG09]

Käytännössä IT:n hallinnointiin pohjaavien päätöksenteon ja vastuuttamisen rakenteiden voidaan nähdä muodostavan vastuunjaon ketjuja organisaatiossa. Henkilö, jolla on jonkin asiakokonaisuuden tulosvastuu (*accountability*), on viime kädessä vastuussa kokonaisuuden hoitamisesta ja raportoisimisesta organisaatiohierarkiassa ylöspäin. Hän ei voi pelkällä omalla päätöksellään delegoida tulosvastuutaan jollekin toiselle. Hän voi kuitenkin jäsentää ja määritellä asiakokonaisuuden tehtäviksi ja delegoida ne yhdelle tai useammalle henkilölle. Vastataanottaessaan tehtävän henkilö hyväksyy sen osalta sekä tulosvastuun että suoritusvastuun (*responsibility*). Vastuunjaon ketju pitenee, mikäli henkilöt jakavat suoritusvastuutaan edelleen seuraaville henkilöille [MGW18, Kal02].

IT-päätöksiä voidaan tehdä muun muassa organisaation johdossa, tietohallinnossa tai ydintoimintojen sisällä. Parhaimmillaan IT:tä koskeva päätöksenteko on kaikkien sidosryhmien välisen yhteistyön tulosta. Päätöksentekoa itsessään voidaan hahmottaa erilaisten kokonaisuuksien kautta. Esimerkiksi Weill ja Ross jäsentävät keskeiset IT:hen liittyvät päätökset viiteen osaan, jotka ovat kiinteästi yhteydessä toisiinsa. Näihin lukeutuvat periaatteet (*IT principles*) eli päätökset IT:n strategisesta roolista, arkkitehtuuri (*IT architecture*) eli tiedon ja järjestelmien integrointia ja vakiointia koskevat päätökset, infrastruktuuri (*IT infrastructure*) eli päätökset organisaation jaetuista ja jatkuvista palveluista, liiketoimintatarpeet (*business application needs*) eli sovelluksiin kohdistuvien liiketoimintatarpeiden määrittelyyn liittyvät päätökset sekä investoinnit ja priorisoinnit (*IT investment and prioritization*) eli päätökset resurssien kohdistamisesta [WeR04].

Pohjimmiltaan päätöksenteossa on kyse valintojen tekemisestä: päätöksentekijä punnitsee vaihtoehtoja, niiden hyötyjä ja haittoja eli kustannuksia, ja tekee lopulta ratkaisun. Tutkimusta on tehty paljon siitä, miten päätöksiä tehdään. Perinteinen lähestymistapa on niin sanottu rationaalisen päätöksenteon malli, jossa oletetaan, että päätöksentekijällä on yksi selkeä tavoite ja kaikki tarvittava tieto eri vaihtoehtoista ja niiden seurauksista. Päätöksillä pyritään siten valitsemaan parhaat tavoitteet ja keinot eli pääsemään toivottuun lopputulokseen mahdollisimman vähäisiä resursseja hyödyntäen. Aina ei kuitenkaan onnistuta valitsemaan parasta vaihtoehtoa: todellisuudessa esimerkiksi hyötyjen ja kustannusten arvioiminen voi

olla hyvinkin hankalaa, minkä lisäksi päätöksentekoon vaikuttavat inhimilliset tekijät [Hyy12].

Kritiikkinä epärealistisena pidetylle rationaaliselle päätöksenteolle on syntynyt erilaisia muita päätöksenteon malleja, kuten rajoittunut rationaalisuus. Siinä päätöksentekijä joutuu tekemään ratkaisunsa rajoitetun käsityksen perusteella. Syynä voi olla esimerkiksi ajan tai rahan kaltaisten resurssien puute tai päätöksentekijän vajavainen ymmärrys päätöksen kohteena olevasta asiasta. Päätöksellä pyritään saavuttamaan tyydyttävä ratkaisu eikä niinkään parasta mahdollista ratkaisua [Sim79, Hyy12]. Vastaavasti pesäeroa rationaaliseen malliin tekee muun muassa inkrementalismi, jossa päätöksenteon ajatellaan tapahtuvan jaksottaisesti ja inkrementaalisesti. Samalla tavoitteet muokkautuvat. Ajatuksena on, että mikäli päätöksentekijä kokee jonkin toimintatavan huonoksi, hän tekee isojen päätösten sijaan pieniä korjauksia tilanteen parantamiseksi [PaW96].

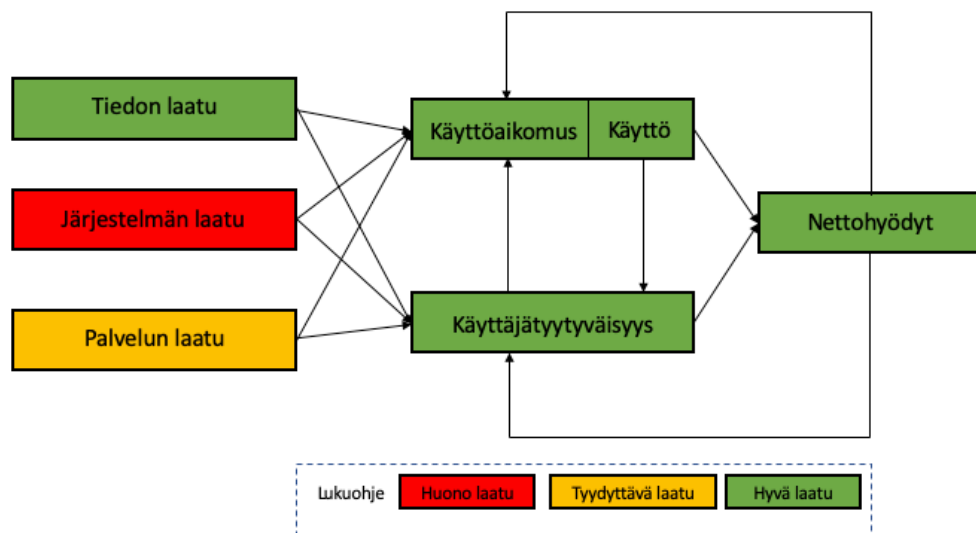
Päätöksenteon suoranaista irrationaalisuutta korostaa esimerkiksi Cohenin, Marchin ja Olsenin organisoitunut anarkia. Tutkijoiden mukaan jokaisessa organisaatiossa on organisoituneen anarkian piirteitä. Tämä tarkoittaa muun muassa sitä, että organisaatioilla on epäselvät toimintatavat ja tavoitteet sekä ongelmallisia arvostuksia. Lisäksi päätöksentekoon osallistuminen on satunnaista ja päätöksentekotilanteissa on usein läsnä eri ihmisiä. Tällaisessa ympäristössä päätöksenteko ei voi olla kovin rationaalista. Ratkaisuihin päästään käsiksi vasta, kun sopiva kokoonpano ongelmia, ratkaisuvaihtoehtoja ja asiasta kiinnostuneita päätöksentekijöitä on olemassa [CMO72].

Mintzberg ja Westley esittävät, että perinteisen rationaalisen ja faktoihin perustuvan päätöksentekoprosessin (*thinking first*) rinnalle on tärkeä sisällyttää oivalluksiin ja ideoihin (*seeing first*) sekä kokemuksiin (*doing first*) perustuvia päätöksentekomalleja [MiW01]. Fredricksonin mukaan päätöksentekoprosessi tarvitsee käynnistyäkseen herätteen (*decision trigger, decision stimulus*). Heräte voi syntyä esimerkiksi reagoitaessa johonkin akuuttiin ongelmaan tai kriisiin tai proaktiivisen toiminnan tuloksena, kuten pyrkimyksestä etsiä uusia liiketoimintamahdollisuuksia. Formaali prosessit voivat toimia lähteinä reaktiiviselle päätöksenteolle muun muassa määrittelemällä aikataulutettuja ja säännönmukaisia päätöksenteon tilanteita. Toisaalta niiden riskinä on, että ne vähentävät ihmisten luovaa, proaktiivista käyttäytymistä [Fre86].

Laajassa kuvassa päätöksentekoa voidaan tarkastella laadun tavoin myös organisaatiokulttuurin kautta. Organisaatiokulttuurilla on yhteys organisaation kykyyn havaita ja nostaa esiin

sellaisia merkityksellisiä ilmiöitä, joihin saattaa liittyä reagoinnin ja päätöksenteon tarvetta [GIK14]. Organisaation kollektiivista päätöksentekokykyä voidaan kehittää vaikuttamalla organisaatiokulttuuriin ja esimerkiksi kannustamalla merkityksellistämiseen (*sensemaking*), jossa takautuvan itsetutkiskelun avulla pyritään ymmärtämään yhteisiä kokemuksia ja liittämään niihin merkityksiä [Wei95]. Esimerkiksi se, mitä organisaatiossa pidetään tärkeänä, miten organisaatiota johdetaan ja minkälaisia toimintatapoja siellä vallitsee, vaikuttaa myös päätöksentekoon.

Tämän luvun lopuksi tarkastellaan vielä yhtä tutkimuksen aiheen kannalta kiinnostavaa näkökulmaa, DeLonen ja McLeanin onnistumisen mallia, johon näkee viitattavan usein tietojärjestelmien johtamista käsittelevässä tutkimuksessa. Mallissa tietojärjestelmän onnistumista hahmotetaan seuraavien muuttujien kautta: järjestelmän laatu, tiedon laatu, palvelun laatu, käyttöaikomus, käyttö, käyttäjäytyvyisyys sekä nettohyödyt. Mallin mukaan tietojärjestelmän, tiedon ja palvelun laatu vaikuttavat saatuun nettohyötyyn mutta epäsuorasti, sidosryhmien kokemusten sekä tietojärjestelmän todellisen käytön kautta [DeM03].



Kuva 3. Havainnollistus tietojärjestelmän onnistumisen mallista [mukaillen DeM03]

Mallin kytköstä päätöksentekoon voidaan hahmottaa seuraavasti: Edellä todettiin, että päätöksenteko on vaihtoehtojen arvioimista ja valitsemista. DeLonen ja McLeanin malli esittää, että tietojärjestelmä voi olla organisaation näkökulmasta onnistunut, vaikka siinä olisi tunnettuja laadullisia puutteita. Kuvassa 3 on tutkijan havainnollistus tästä. Toisin sanoen malli selittää osaltaan sellaisia päätöksentekotilanteita, joissa korkeaa laatua ei välttämättä valita keskeiseksi tavoitteeksi, vaan siitä saatetaan tietoisesti tinkiä nettohyötyjen ollessa riittävän

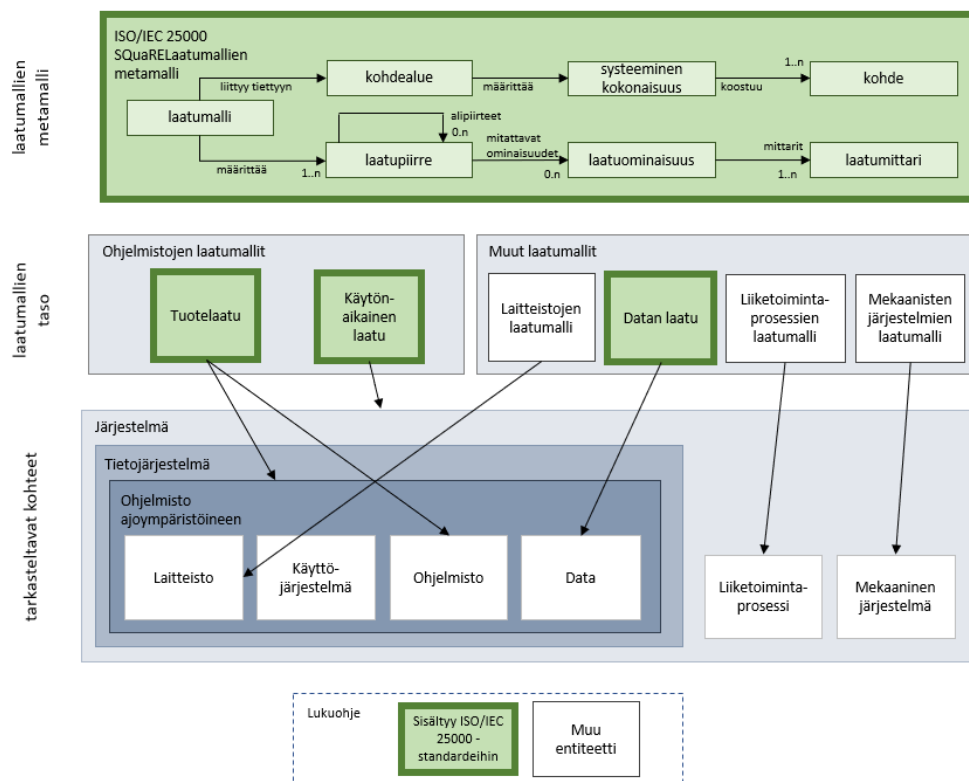
suuria. Onnistumisen maksimoimiseksi päätöksenteossa on hyvä ymmärtää riittävällä tasolla, missä määrin laadussa voidaan joustaa, sekä tuntee mahdollisia muita laatuun vaikuttavia ilmiöitä, joita kartoitetaan tässä tutkimuksessa.

2.3 Ohjelmistotuotannon viitekehyksiä

Seuraavaksi esitellään valikoitu joukko toimialalla yleisesti tunnettuja, laajalti käytettyjä ja avoimia viitekehyksiä ja standardeja. Ne ovat luonteeltaan toisiaan täydentäviä, ja niiden pohjalta organisaatio voi koostaa oman IT:n hallinnoinnin, johtamisen ja ohjelmistotuotannon menetelmästänsä. Viitekehysten merkitys tutkielmalle on toimia aineistona, jota vasten peilataan haastatteluissa esiin nousevia ilmiöitä. Kukin viitekehys tai standardi ja sen käyttötarkoitus kuvataan yleisellä tasolla.

2.3.1 ISO/IEC 25000 -laatustandardisarja SQuaRE

ISO/IEC 25000 eli System and Software Quality Requirements and Evaluation (SQuaRE) on kansainvälinen laatustandardisarja, joka ottaa kantaa ohjelmistotuotteiden laatuun. Se määrittää laatumalleille metatason käsitelmän, johon pohjaavat seuraavat kolme laatumallia: tuotelaadun malli, käytönaikaisen laadun malli sekä datan laatumalli (kuva 4) [ISO11a, ISO11b].



Kuva 4. Laatustandardisarjan ISO/IEC 25000 laatumallit [mukaillen ISO11a ja ISO11c]

Jokainen metamallia noudattava laatumalli määrittää joukon laatupiirteitä (*characteristics*). Laatupiirteet voivat jakautua alipiirteisiin (*subcharacteristics*), jotka puolestaan voivat edelleen jakautua pienemmiksi kokonaisuuksiksi. Jos laatupiiirre ei ole suoraan mitattavissa, sillä on yksi tai useampi laatumittareilla mitattavissa oleva laatuominaisuus. Laatumalli liittyy aina tiettyyn kohdealueeseen eli pääasialliseen käytön kontekstiin. Sen kautta määritetty tarkastelun kohteena oleva systeeminen kokonaisuus, joka koostuu yhdestä tai useammasta kohteesta. Taulukko 1 esittää esimerkin vuoksi tuotelaadun laatupiirteiden ja alipiirteiden hierarkian.

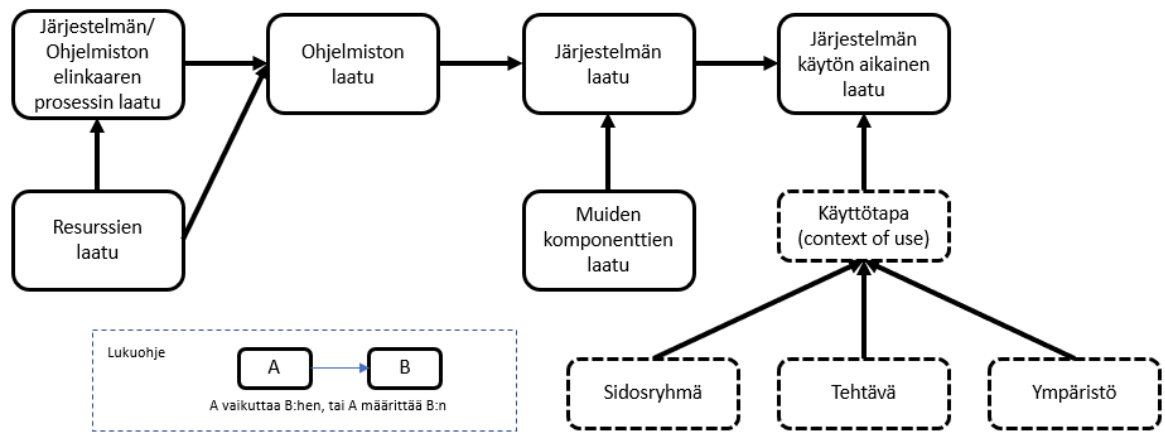
Toiminnallinen soveltuvuus	Suorituskyky	Yhteensopivuus	Käytettävyys
<ul style="list-style-type: none"> Toiminnallinen kattavuus Toiminnallinen oikeellisuus Toimintojen oikeanlainen suunnittelu ja toteutus 	<ul style="list-style-type: none"> Aikakäyttäytyminen Resurssien käyttö Kapasiteetti 	<ul style="list-style-type: none"> Yhteiselo toisten järjestelmien kanssa Yhteentoimivuus 	<ul style="list-style-type: none"> Tarkoitukseen soveltuvuuden tunnistaminen Opittavuus Operoitavuus Suojaus käyttäjän virheiltä Käyttöliittymän esteettisyys Saavutettavuus
Luotettavuus	Tietoturvallisuus	Ylläpidettävyys	Siirrettävyys
<ul style="list-style-type: none"> Kypsyy Saatavuus Vikasietoisuus Toipumiskyky 	<ul style="list-style-type: none"> Luottamuksellisuus Eheys Kiistämättömyys Tarkastettavuus Autentikointi 	<ul style="list-style-type: none"> Modulaarisuus Uudelleenkäytettävyys Analysoitavuus Muunnettavuus Testattavuus 	<ul style="list-style-type: none"> Sovitettavuus Asennettavuus Korvattavuus

Laatupiiirre
<ul style="list-style-type: none"> Alipiirre

Taulukko 1. Ohjelmiston tuotelaadun mallin mukaiset laatupiirteet ja alipiirteet
[mukaillen ISO11a]

Ohjelmistotuotteen laatuun vaikuttavat resurssit, ohjelmistotuotantoprosessit, tietojärjestelmän muiden komponenttien laatu sekä käytön konteksti, joka muodostuu sidosryhmistä, käyttötapauksista ja ympäristöstä (kuva 5, sivu 14). Tietojärjestelmällä voi esimerkiksi olla erilaisia käyttäjäryhmiä käyttötapauksineen, tarpeineen ja odotuksineen, ja ryhmien kokemukset tietojärjestelmän laadusta saattavat poiketa toisistaan [ISO11a].

Standardia voidaan hyödyntää esimerkiksi vaatimusmäärittelyssä, laatuarvioinnissa sekä ohjelmistometriikan mittarien käyttöönotossa. Malleissa kuvattuja laatupiirteiden koonteja voidaan lisäksi käyttää tarkistuslistoina varmistamaan, että laatuvaatimukset tulee käytyä kattavasti läpi [ISO11a].



Kuva 5. Tuotelaatuun vaikuttavia tekijöitä [mukaillen ISO11a]

2.3.2 COBIT

1960-luvun loppupuolella joukko tilintarkastuksen ja sisäisen tarkastuksen ammattilaisia perusti työnsä tueksi ja tietämyksen jakamiseksi yhdistyksen EDP Auditors Association [ISA20]. EDP on lyhenne sanoista Electronic Data Processing [GSM04]. Yhdistyksen perustamisen taustalla oli havainto siitä, että tarkastajat olivat alkaneet kohdata työssään yhä enemmän automaattista tietojenkäsittelyä ja kriittisissä rooleissa toimivia tietojärjestelmiä. Tarvittiin kanava, jonka kautta oli mahdollista jakaa keskitetysti työhön liittyvää tietoa ja ymmärrystä [ISA20]. Nykyään yhdistyksen nimi on Information Systems Audit and Control Association (ISACA). Sillä on yli 165000 jäsentä 180:ssä maassa [ISA20] sekä maakohtaisia jäsenjärjestöjä kuten ISACA Finland Chapter Tietojärjestelmien tarkastus ja valvonta ry. Yhdistyksen tavoitteena on hyvän tietohallintotavan kehittäminen ja edistäminen [ISA19].

Control Objectives for Information and Related Technology (COBIT) on ISACAn julkaisema IT:n hallinnoinnin ja johtamisen parhaiden käytäntöjen viitekehys [COB18a]. Sen ensimmäinen versio ilmestyi 1990-luvun puolivälissä. Tässä tutkimuksessa tarkastellaan COBITin uusinta versiota COBIT 2019. Versioiden myötä COBIT on laajentunut tarkastusprosesseja ohjaavasta viitekehyksestä kokonaisvaltaisemmaksi IT:n johtamisen ja hallinnoinnin viitekehykseksi [BSH15].

COBITin keskeisenä ajatuksena on rakentaa polku organisaation strategiatason tarpeista ja niiden taustalla olevista ajureista toteutuskelpoiseen (*actionable*) strategiaan (kuva 6, s. 15). Polun käsitteiden välillä on monesta-moneen-yhteydet, joten tarkalleen ottaen kyse on riippuvuuksien muodostamasta verkostosta. Kuljettaessa polkua eteenpäin (kuvassa alaspäin) siirrytään liiketoiminnan ja IT:n yhteensovittamisen alueelle. Tarpeet kuvataan ensin orga-

The diagram illustrates the relationship between organizational goals, strategy, and design factors. It shows a flow from 'Ajurit' (Drivers) through 'Sidosryhmien tarpeet, odotukset ja vaatimukset' (Stakeholder needs, expectations and requirements) to 'Yritystason päämäärät' (Organizational goals). These goals lead to 'Liiketoiminnan ja IT:n yhteensovittamisen päämäärät' (Business and IT integration goals), which then lead to 'Hallinnon tavoitteet' (Management goals) and 'Johtamisen tavoitteet' (Leadership goals). These goals are influenced by 'Mittarit' (Metrics) and 'Kohdealueet' (Target areas). The final outcome is 'Organisaation komponentit, strategiatyön syötteenä ja muutoskohteena' (Organizational components, as input and target for strategy work). This leads to a table of components and their goals, which then leads to 'ISO/IEC, PMBOK, ...' (Standards, frameworks, etc.).

Organisaation komponentit, strategiatyön syötteenä ja muutoskohteena	Yhteys tavoitteisiin
<p>Prosessit</p> <p>Organisaatorakenne</p> <p>(johtamisen) tietovirrat ja tieto</p> <p>Henkilöstö, taidot, kompetenssit</p> <p>Periaatteet, käytännöt, proseduurit</p> <p>Kulttuuri, etiikka, toimintatavat</p> <p>Palvelut, infrastruktuuri, sovellukset</p>	<p>Käytännöt ja aktiviteetit, kyvykkyyssoihin jäsennettyinä</p> <p>Käytäntöjen ja aktiviteettien vastuunjakko (RACI)</p> <p>Tiedon kulku: lähteet ja kohteet (yleensä: prosessit)</p> <p>Tavoitteen saavuttamiseksi tarvittavat henkilöresurssit ja taidot</p> <p>Viittaus oleellisiin käytäntöihin ja proseduureihin</p> <p>Viittaus keskeisiin kulttuurillisiin elementteihin</p> <p>Viittaus sovellus-/palvelutyyppiin joka voi olla tarpeellinen tavoitteen saavuttamisessa. Esimerkki: "Riskienhallintajärjestelmä."</p>

viittaus lisäohjeisiin (standardit, viitekehykset)

ISO/IEC, PMBOK, ...

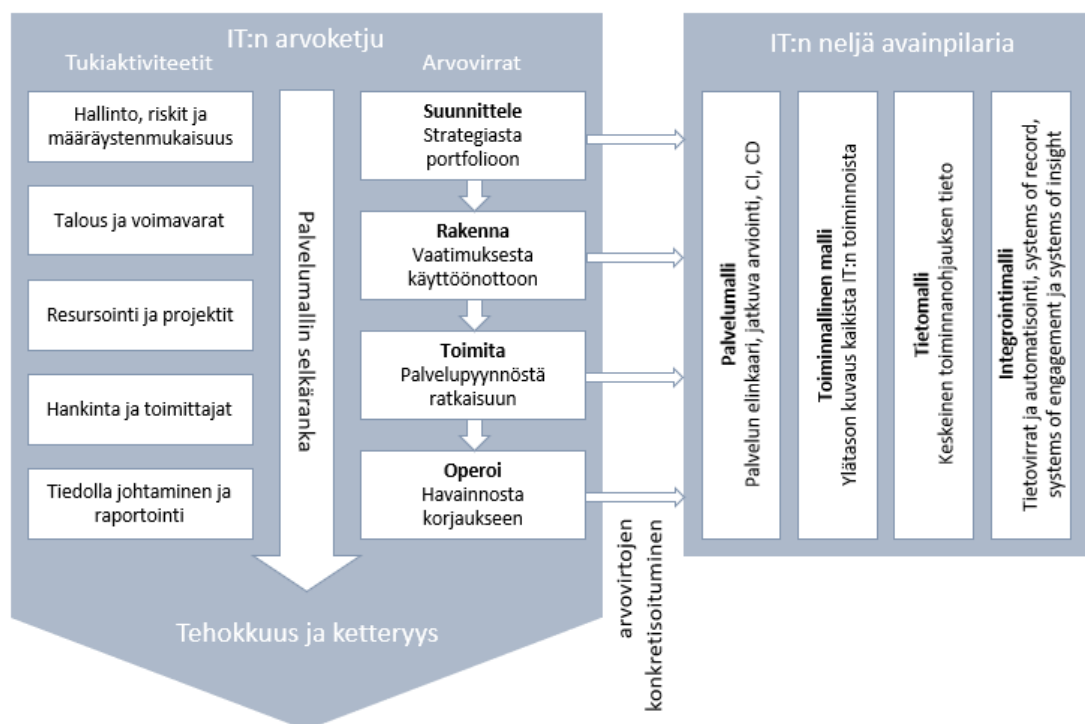
COBITin terminologiassa viitekehyksen (*governance framework*) käyttöönotto tarkoittaa käytännössä sitä, että organisaatio räätälöi itselleen konkreettisen hallintojärjestelmän COBITin avulla. Viitekehys määrittelee räätälöintiprosessia varten kaksi keskeistä käsitettä: organisaation hallintojärjestelmän suunnittelun tekijät (*design factors*) sekä organisaation kohdealueet.

IT4IT™ Reference Architecture on yksi The Open Groupin standardeista, avoin ja puolueeton viitearkkitehtuuri organisaation IT-toiminnon organisoimiseen ja johtamiseen [IT417].

26.10.2020 sen käyttäjäfoorumin jäsenlistalla oli 102 organisaatiota [IT420]. Mallin ensimmäinen versio julkaistiin vuonna 2015 [TaF18]. Tässä tutkimuksessa tarkastellaan IT4IT-mallin tuoreinta versiota 2.1 vuodelta 2017.

IT4IT-viitearkkitehtuurissa teknologiajohtamista sekä liiketoiminnan ja IT:n yhteensovittamista tarkastellaan arvoketjumallin kautta. Arvoketju on Harvardin yliopiston tutkija Michael Porterin kehittämä malli yrityksen arvonmuodostusprosessista [Por85]. IT4IT-arkkitehtuuri määrittelee IT:n arvoketjun (*IT value chain*) sarjaksi aktiviteetteja, joiden kautta IT:n tuottama palvelu tai tuote kulkee elinkaarensa aikana [IT417].

IT:n arvoketju muodostuu neljästä toiminnallisesta komponentista – arvovirrasta – ja viidestä tukiaktiviteetistä (kuva 7). Arvovirrat ovat nimeltään suunnittele eli strategiasta portfolioon, rakenna eli vaatimuksesta käyttöönottoon, toimita eli palvelupyynnöstä ratkaisuun sekä operoi eli havainnosta korjaukseen. Tukiaktiviteetit ovat nimeltään hallinto, riskit ja määräystenmukaisuus; talous ja voimavarat; resursointi ja projektit; hankinta ja toimittajat sekä tiedolla johtaminen ja raportointi [IT417].



Kuva 7. IT4IT-viitearkkitehtuurin arvoketju ja neljä peruspilaria [mukaillen IT417]

Arvovirtojen ja arvoketjujen konkretisoituminen organisaatiossa tapahtuu IT:n neljän avainpilarin kautta (kuva 7, s. 17), jotka ovat palvelumalli, toiminnallinen malli, tietomalli ja integrointimalli. Ne eivät ota kantaa teknologioihin, prosesseihin, menetelmiin ja kyvykkyyksiin, vaan sen sijaan mallintavat asioita, joiden oletetaan pysyvän muuttumattomampina, kuten tietoa, toiminnallisia komponentteja ja komponenttien läpi kulkevia tietovirtoja. Toiminnallinen komponentti tarkoittaa prosessimaista palvelun jalostusvaihetta, kuten versionhallintaa, sovellusten paketointia ja sovellusten asennusta. IT4IT pyrkii edesauttamaan IT:n palvelutuotannon tehostamista automatisoinnilla ja työkaluilla [IT417].

IT4IT:tä käytettäessä uuden palvelun tai tuotteen tuottaminen tai olemassa olevan kehittäminen tapahtuu iteratiivisesti ja inkrementaalisesti kulkemalla arvovirtojen vaiheiden läpi. Viitearkkitehtuuri ei pakota ajamaan jokaista muutosta kaikkien arvovirtojen läpi vesiputousmaisesti, vaan aiempien arvovirtojen tuotoksia voidaan käyttää uudelleen. Esimerkiksi joissakin tilanteissa muutoksia saatetaan ajaa suoraan toimita-arvovirran läpi [IT417].

Arvovirrat kytkevät yhteen toimintoja, jotka ovat aikaisemmin saattaneet olla erillään ja joista on mahdollisesti ollut vaikea muodostaa organisaatiotason kokonaiskuvaa. IT4IT:n tavoitteena on pyrkiä muodostamaan kokonaisvaltaisempi ymmärrys IT:n toiminnasta ja nostaa IT:n liiketoiminnalle tuoma arvonlisäys selkeämmin esille [IT417]. IT4IT voidaan nähdä keinoksi toteuttaa DevOps-ajattelutavan mukaista organisoitumista [TOG19].

2.3.4 TOGAF

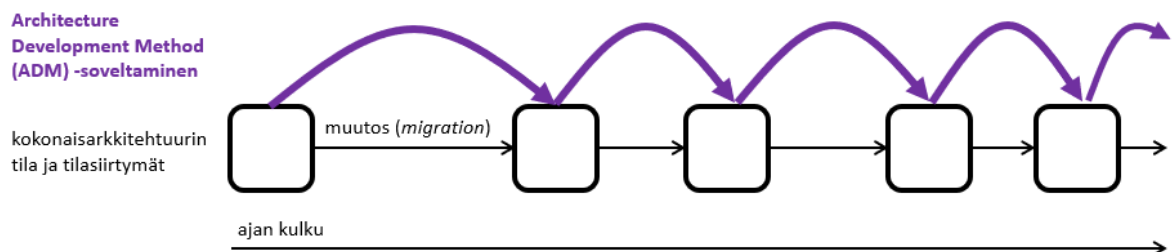
The Open Group on maailmanlaajuinen, yli 790 jäsenorganisaation muodostama yhteenliittymä, jonka eräs toiminnan muoto on määritellä ja julkaista avoimia IT-toimialan standardeja. TOGAF® Standard, kokonaisarkkitehtuurin tai yritysarkkitehtuurin (*enterprise architecture*) viitekehys, on yksi The Open Groupin julkaisemista standardeista [TOG18].

Virallinen TOGAF-dokumentaatio jakautuu kahteen osaan: yhtenäisenä dokumenttina julkaistuun standardiin sekä TOGAF® Libraryyn, joka tarjoaa standardia tukevaa dokumentaatiota, kuten terminologian käännöksiä eri kielille, opiskelumateriaalia, webinaareja ja tiiviitä tietokortteja. TOGAF® Series Guides on Libraryn kautta julkaistu dokumenttipaketti, joka sisältää yksityiskohtaisempia ohjeita standardin käyttämiseksi. Tässä tutkimuksessa tarkastellaan pääasiassa standardia ja sen tuoreinta TOGAFin versiota 9.2, joka on julkaistu vuonna 2018. TOGAF Series Guides -dokumenttipaketin (14 dokumenttia) tuorein versio

huhtikuulta 2020 [TOG20] on sisällytetty mukaan tarkistettaessa, mitä haastatteluissa esiin nousevia käsitteitä TOGAF eksplisiittisesti mainitsee.

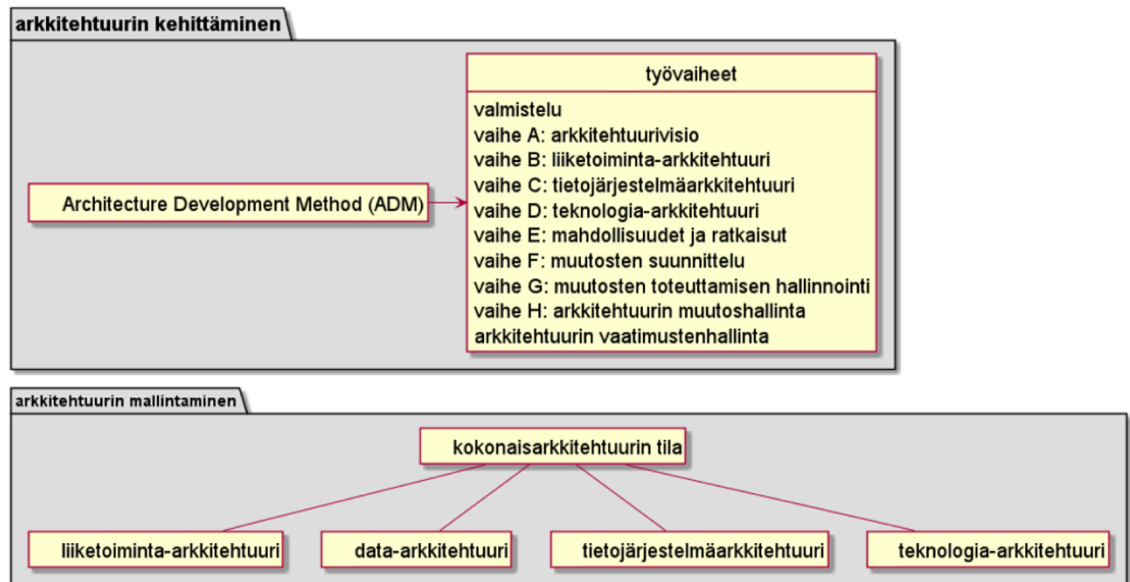
TOGAF määrittelee itsensä parhaiden käytäntöjen viitekehykseksi, joka yhdessä TOGAF Libraryn käytännön sovellutusta tukevan dokumentaation kanssa tarjoaa menetelmiä ja työkaluja organisaatioiden kokonaisarkkitehtuurityölle [TOG18]. Organisaatiolla on aina olemassa jonkinlainen kokonaisarkkitehtuuri, joka voi olla suunniteltu tai suunnittelematon. Kokonaisarkkitehtuurityötä tukeva viitekehys kuten TOGAF astuu kuvaan, kun kokonaisarkkitehtuuria halutaan systemaattisesti suunnitella, ohjata ja kehittää koko organisaation elinkaaren ajan.

Ylätasolla TOGAF hahmottaa kokonaisarkkitehtuurin tilan mallintamisen ja muutosten eli tilasiirtymien (*migration*) suunnittelun kautta (kuva 8). Kokonaisarkkitehtuurin kehittäminen perustuu voimassaolevan tilan mallinnukseen ja tavoitetilan määrittelyyn sekä tilasiirtymien ohjaamiseen TOGAFin Architecture Development Method (ADM) -menetelmällä. ADM-menetelmä määrittelee kymmenen vaihetta (kuva 9, s. 19), joita toistetaan iteratiivisesti organisaation elinkaaren aikana kokonaisarkkitehtuurin jatkuvaksi kehittämiseksi [TOG18].



Kuva 8. Kokonaisarkkitehtuurin tilan mallinnus ja tilasiirtymien ohjaus TOGAF-viitekehyksessä [TOG18]

Kokonaisarkkitehtuurin tilaa mallinnetaan neljän näkökulman kautta (kuva 9, s. 19). Liiketoiminta-arkkitehtuuri (*business architecture*) määrittelee liiketoiminnan strategian, hallinnon, organisaatorakenteen ja keskeiset liiketoimintaprosessit. Data-arkkitehtuuri (*data architecture*) kuvaa organisaation datapääomaa ja tietovirtoja. Tietojärjestelmäarkkitehtuuri (*application architecture*) muodostuu organisaation kaikista tietojärjestelmistä ja niiden välisistä sidoksista. Teknologia-arkkitehtuuri (*technology architecture*) kuvaa teknologiaa, jota tarvitaan yrityksen tietojärjestelmien ja tietovirtojen perustaksi [TOG18].

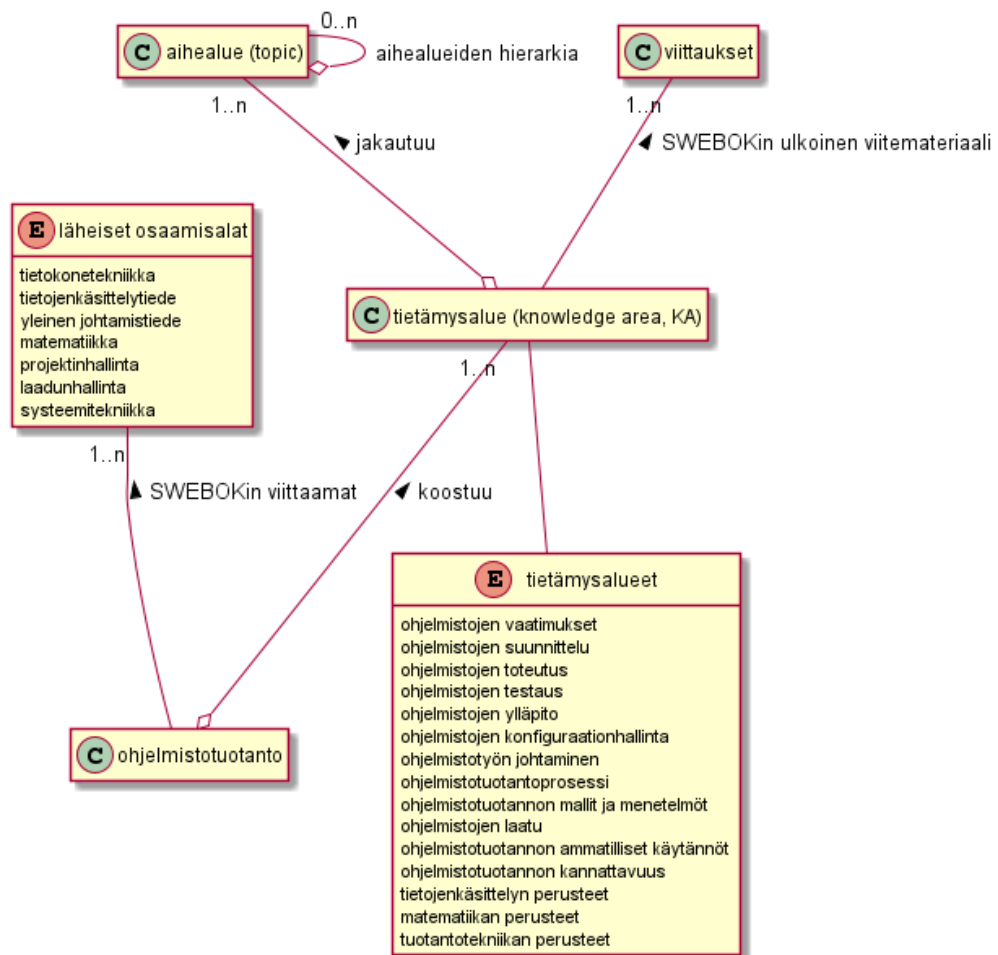


Kuva 9. Kokonaisarkkitehtuurin mallintamisen ja kehittämisen ylätasoon käsitteet TOGAF-viitekehyksessä [TOG18]

2.3.5 SWEBOK

Guide to the software engineering body of knowledge (SWEBOK®) [SWE14] on IEEE Computer Societyn julkaisema vapaasti saatavilla oleva käsikirja sekä kansainvälinen standardi ISO/IEC TR 19759:2015 [ISO15]. Se valikoi, organisoi ja määrittelee kaikesta ohjelmistotekniikan tietämyksestä keskeiset ja yleisesti tunnistetut ja hyväksytyt osa-alueet. Sen tavoitteina on muun muassa edistää maailmanlaajuisesti yhtenäistä käsitystä ohjelmistotuotannon osaamisalasta, määritellä alan sisältöä, laajuutta ja sijoittumista läheisiin osaamisaloihin sekä toimia pohjana alan opetussuunnitelmien kehitykselle. SWEBOKin ensimmäinen versio julkaistiin vuonna 2004 osana pyrkimystä määritellä ohjelmistotuotannon osaamisalaa. Tässä tutkimuksessa tarkastellaan SWEBOK-käsikirjan tuoreinta versiota 3.0, joka on julkaistu vuonna 2014 [SWE20, SWE14].

SWEBOKin metamallissa ylimmän tason käsite on tietämysalue (*knowledge area*, KA), joita on 15 kappaletta SWEBOKin versiossa 3.0. Kukin tietämysalue jakautuu aihealueisiin (*topic*), jotka voivat muodostaa hierarkian jakautumalla edelleen tarkempiin aihealueisiin (kuva 10, s. 20). SWEBOKin luonteenomaisena piirteenä on viitemateriaalin esittäminen tietämysalueittain matriiseissa, joiden avulla lukija saa visualisoidun yleiskuvan siitä, mitkä lähteet ovat SWEBOKin näkökulmasta alan perusteoksia tai muutoin keskeisiä ja mitkä aihealueet viittaavat mihinkin lähteeseen [SWE14].



Kuva 10. Keskeiset metatason käsitteet SWEBOK v3.0 -käsikirjassa [SWE14]

2.3.6 PMBOK

Project Management Institute (PMI) on voittoa tavoittelematon projektinhallinnan ammattijärjestö. Järjestö perustettiin vuonna 1969. Aluksi sen tarkoituksena oli toimia kanavana kokemusten jakamiselle projektinhallinnan parissa työskenteleville. Sitten yhdistyksen tarkoitus on laajentunut, ja nykyään PMI:n pyrkimyksenä on edistää projektinhallinnan käytännön tietoutta projektinhallinnan ammattilaisten keskuudessa muun muassa julkaisemalla kirjallisuutta ja tarjoamalla sertifiointipalveluja [Sli08, PMI20].

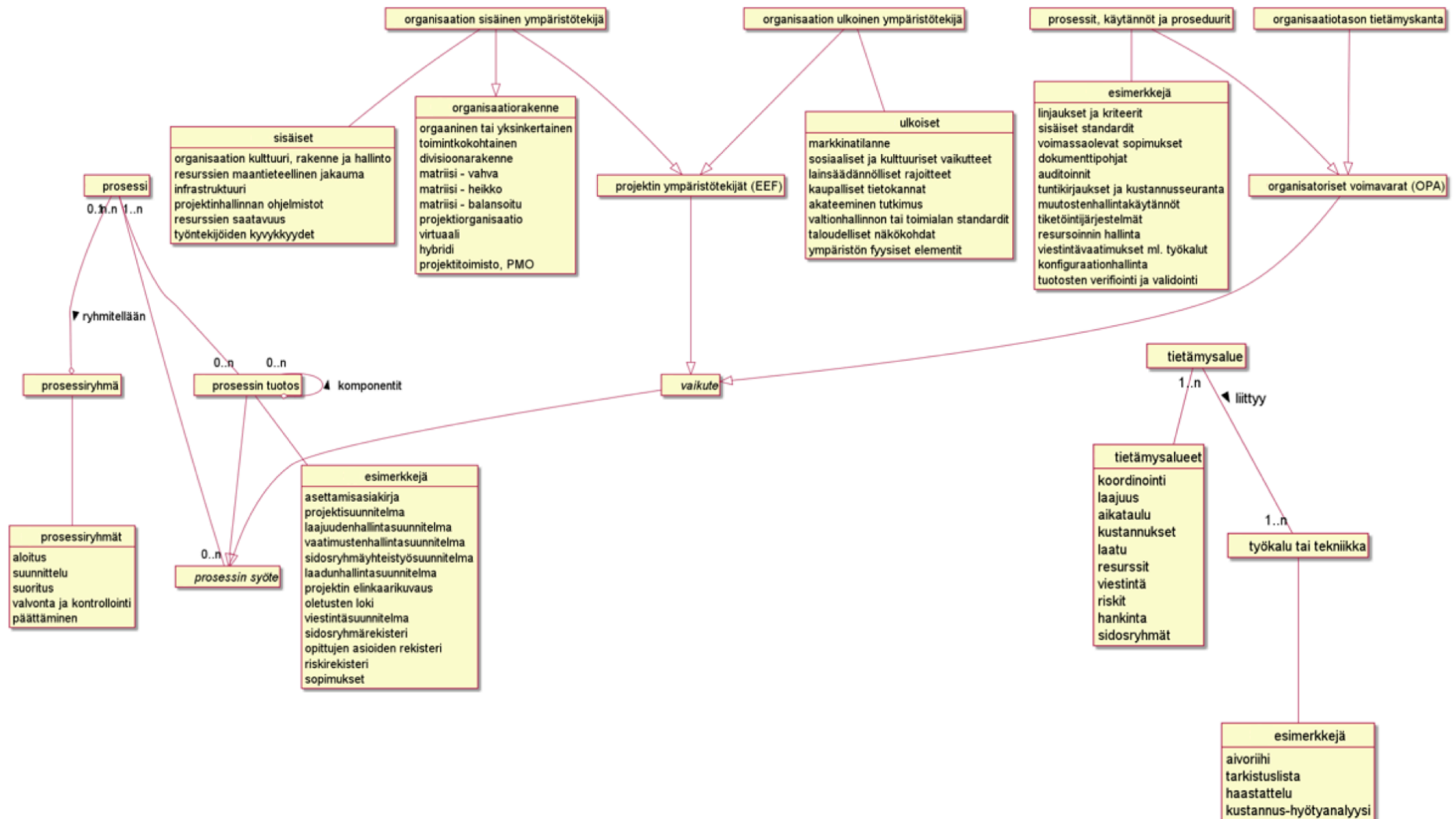
Project Management Body of Knowledge (PMBOK®) on PMI:n määrittelemä termi, jolla tarkoitetaan kaikkea projektinhallinnan ammattialaan liittyvää, jatkuvasti muuttuvaa ja kehittyvää, julkaistua ja julkaisematonta tietämystä [PMB17]. The Guide to the PMBOK (PMBOK-käsikirja) on PMI:n julkaisema dokumentti, jonka tavoitteena on kuvata PMBOK:n kaikesta mahdollisesta sisällöstä yleisesti hyvinä pidetyt ja tunnetut prosessit ja

terminologia. Tässä tutkimuksessa tarkastellaan PMBOK-käsikirjan tuoreinta julkaisua, joka on kuudes versio vuodelta 2017 [PMB17].

PMBOK-käsikirja on jäsennetty liitteiden lisäksi kahteen osaan. Jälkimmäinen on noin sataisivuinen projektinhallinnan standardi ANSI/PMI 99-001-2017. Se määrittelee prosesseja, joita voidaan pitää hyvinä käytäntöinä suurimmalle osalle projekteista suurimman osan ajasta. Standardi toimii samalla viitekehyksenä ja perustana dokumentin ensimmäiselle osalle, joka on varsinainen käsikirjaosio ja kooltaan noin viisisataa sivua. Käsikirjaosuus syventää kuvausta projektien kontekstista ja vaikutteista, määrittelee projektipäällikön roolin, prosessien syötteet ja tuotokset sekä projektinhallinnan työkaluja ja tekniikoita. Lisäksi se esittelee kuhunkin osa-alueeseen liittyviä avainkäsitteitä ja trendejä [PMB17].

Keskeinen metatason käsite PMBOK-käsikirjassa on prosessi (*process*), jolla mallinnetaan projektien työvaiheita. Prosessit on ryhmitelty prosessiryhmiin. Prosessit ottavat syötteitä ja tuottavat tuotoksia. Syötteet ovat paitsi aiempien prosessien tuotoksia, myös projektin ulkoisia vaikutteita (*influence*). Vaikutteet luokitellaan projektin ympäristötekijöihin (*enterprise environmental factor*, EEF) ja organisatorisiin voimavaroihin (*organizational process assets*, OPA) (kuva 11, s. 22). Projektin ympäristötekijät ovat organisaation sisältä tai ulkopuolelta tulevia vaikutteita, joihin lähtökohtaisesti ei voida vaikuttaa projektista käsin. Organisatoriset voimavarat puolestaan muodostuvat organisaation käytössä olevista prosesseista, käytännöistä, työkaluista ja tietämyksestä [PMB17].

Luonteeltaan PMBOK-käsikirja on kuvaileva. Se ei kerro tarkasti esimerkiksi, miten projekti tulee viedä läpi, mitä asioita tehdään missäkin järjestyksessä tai miten projektitiimi organisoituu ja jakaa vastuunsa. Sitä vastoin sen esittelemien käytäntöjen ja välineiden pohjalta organisaatiot voivat räätälöidä omat menetelmänsä projekteja varten [PMB17]. Käsikirjaa voi käyttää vaikkapa ketterien menetelmien kanssa [Sli08].



Kuva 11. PMBOK-käsikirjan metamallin keskeiset käsitteet ja niiden väliset suhteet [PMB17]

3 Menetelmät

Tutkimusmenetelmillä tarkoitetaan tutkimuksessa hyödynnettäviä aineiston hankinta-, analyysi- ja tulkintamenetelmiä ja -tekniikoita [Kan08]. Seuraavissa alaluvuissa esitellään tässä tutkimuksessa tehdyt menetelmälliset valinnat ja tutkimuksen toteutus. Aluksi tarkastellaan laadullista tutkimusotetta, joka voidaan mieltää eräänlaiseksi laajaksi lähestymistavaksi tutkimuksen tekemiseen. Tämän jälkeen kerrotaan aineiston keräämisestä teemahaastatteluilla – haastateltavien valinnasta ja haastattelujen tekemisestä, valituista teemoista ja niiden käsitteystä – sekä sisällönanalyysistä aineiston analysointimenetelmänä.

3.1 Laadullinen tutkimusote

Tutkimusmenetelmät jaotellaan usein laadullisiin eli kvalitatiivisiin ja määrällisiin eli kvantitatiivisiin menetelmiin. Vanhastaan tutkimusotteet ovat näyttäneet kilpailevina lähestymistapoina tutkimuksen tekemiseen. Todellisuudessa erottelu on varsin karkea, eivätkä menetelmät ole toisiaan poissulkevia [HiH04]. Vertailun avulla voidaan kuitenkin tuoda selkeyttä valitun tutkimusotteen – tässä tapauksessa laadullisen – ominaispiirteisiin. Taulukossa 2 on esitetty joitakin laadullisen ja määrällisen tutkimuksen eroja.

	Laadullinen tutkimus	Määrällinen tutkimus
Tarkoitus	Tulkinta Ymmärtäminen	Yleistäminen Ennustaminen Syy–seuraussuhteet
Lähestymistapa	Päätyy hypoteeseihin ja teorioihin	Alkaa hypoteeseista ja teorioista
Päätelyn logiikka	Induktiivinen (käytännöstä teoriaan)	Deduktiivinen (teoriasta käytäntöön)
Tieto	Tekstit, sanat	Luvut
Tiedon luonne	Subjektiiivinen	Objektiivinen
Tutkijan rooli	Ymmärtäjä Osallistuja	Objektiivinen tutkija Puolueeton Ulkopuolinen

Taulukko 2. Laadullisen ja määrällisen tutkimuksen eroja [mukaiillen Kan08]

Siinä missä määrällisessä tutkimuksessa pyritään tavallisesti syy–seuraussuhteiden hahmotamiseen ja ilmiöiden selittämiseen numeerisesti [Kop15], laadullisen tutkimuksen ominaispiirteeksi mainitaan pyrkimys ymmärtää kokonaisvaltaisesti ja syvällisesti tutkimuskoh-

detta. Kiinnostus suuntautuu lukujen sijasta sanoihin ja teksteihin, joista etsitään merkityksiä, vihjeitä siitä, miten ihmiset kokevat todellisuutensa [Kan08]. Tavoitteena ei ole niinkään löytää totuutta tutkittavasta ilmiöstä, vaan nostaa tulkintojen avulla esille asioita, jotka ovat ”välittömän havainnon tavoittamattomissa” [Vil15]. Pyrkimys ymmärtää ilmiöitä ihmisten kokemusten kautta oli peruste sille, miksi laadullinen tutkimusote valittiin myös tämän tutkimuksen lähestymistavaksi.

Eskolan ja Suorannan mukaan laadullisessa tutkimuksessa tutkimuskohteena on monesti varsin rajallinen määrä tapauksia, joita analysoidaan perusteellisesti. Näin ollen tutkimuksen tieteellisyys perustuu laatuun, ei määrään. Erona määrälliseen tutkimukseen tutkimushypoteeseja ei yleensä aseteta, vaan vahvojen ennakko-oletusten sijaan tutkija pyrkii yllättymään ja oppimaan tutkimuksen edetessä [EsS00]. Kananen mainitsee, että laadulliselle tutkimukselle on kuvaavaa pikemminkin hypoteesien keksiminen kuin niiden testaaminen. Lisäksi tutkija tarkastelee rikasta aineistoaan usein induktiivisesti eli aineistolähtöisesti, ilman teoreettista viitekehystä. Sen sijaan määrällisessä tutkimuksessa päättelyn logiikka perustuu deduktioon: edetään teoriasta käytäntöön, laajemmista kokonaisuuksista pienempiin osiin [Kan08].

Myös tutkijan rooli ja suhde tutkittaviin on erilainen laadullisessa ja määrällisessä tutkimuksessa. Viimeksi mainitulle on ominaista tutkijan objektiivisuus ja etäisyyden säilyttäminen informantteihin [Kan08]. Laadulliseen tutkimukseen liittyy sitä vastoin osallistuvuuden ajatus: tutkija on eräänlainen sisäpiirin tarkkailija [Hak15], joka on vuorovaikutuksessa tutkitavien kanssa ja yrittää tavoittaa heidän näkökulmansa [EsS00]. Laadullinen tutkimus on siten väistämättä subjektiivista tutkimusta. Tämä ei kuitenkaan tarkoita, etteikö yleinen objektiivisuuden vaatimus – omien uskomusten, asenteiden ja arvostusten erillään pitäminen tutkimuskohteesta – pätsisi myös laadulliseen tutkimukseen [Kan08]. Objektiivisuuden voidaan nähdä syntyvän oman subjektiivisuuden tiedostamisesta ja esille tuomisesta [EsS00].

Mitä tulee tutkimusprosessiin, laadulliselle tutkimukselle on ominaista, että sen eri vaiheet – kuten aineistonkeruu, analysointi, tulkinta ja raportointi – nivoutuvat yhteen [EsS00]. Tutkimusprosessin jokainen vaihe vaikuttaa sekä muihin käynnissä oleviin että tuleviin vaiheisiin, ja vaiheet voivat toteutua myös samanaikaisesti [Kan08]. Esimerkiksi tulkintaa tehdään luontevasti jo aineistonkeruun aikana, mikä edelleen ohjaa aineistonkeruun seuraavia askeleita [Hak15]. Joskus on tarpeen palata taaksepäin ja jopa tarkistaa tutkimuksen kysymyk-

senasettelua [EsS00]. Kananen luonnehtii laadullisen tutkimuksen tekemisen mallia sykliseksi erotuksena määrällisen tutkimuksen lineaariseen etenemistapaan. Laadullinen tutkimus elää ja edellyttää tutkijalta jatkuvaa reflektointia [Kan08].

3.2 Aineistonkeruu teemahaastatteluilla

Tyypillisiä aineistonkeruumenetelmiä laadullisessa tutkimuksessa ovat haastattelut, kyselyt, havainnointi ja erilaisten dokumenttien hyödyntäminen [TuS03]. Suomessa kenties käytetyin menetelmä on haastattelu [EsS00], jonka etuna on muun muassa joustavuus: tutkija voi käydä keskustelua haastateltavan kanssa, kysellä, toistaa ja selventää asioita [TuS03]. Kyse on päämäärähakuisesta, suunnitelmallisesta informaation keräämisestä [HiH04], joka mahdollistaa syvällisen tiedon hankkimisen tutkimuksen kohteesta melko lyhyessä ajassa. Tämä tekee haastattelusta otollisen menetelmän esimerkiksi sellaisiin tilanteisiin, joissa aihe on vähän tutkittu tai sitä halutaan ymmärtää paremmin [OMR15].

Haastattelut voidaan toteuttaa eri tavoin. Tavallisesti haastattelumenetelmät jaotellaan niiden strukturointiasteen perusteella strukturoituihin eli lomakehaastatteluihin, puolistrukturoituihin haastatteluihin ja strukturoimattomiin eli avoimiin haastatteluihin [HiH04]. Käytännössä haastattelumenetelmät eroavat siinä, kuinka paljon ne sallivat tutkijalle liikkumavaraa haastattelutilanteessa ja kuinka tarkasti kysymykset ja haastattelu on ylipäänsä suunniteltu [OMR15].

Tämän tutkimuksen aineistonkeruumenetelmäksi valittiin teemahaastattelu, joka voidaan mieltää yhdeksi puolistrukturoidun haastattelun muodoksi [HiH04]. Teemahaastattelussa haastateltavien kanssa käsiteltävät aihekokonaisuudet eli teemat on määritelty etukäteen, mutta niiden käsittelyjärjestys ja laajuus vaihtelevat haastattelujen välillä [EsS00]. Valintaa puolsi teemahaastattelulle ominainen piirre, keskustelevuus, joka auttaa nostamaan esiin asioiden moniselitteisyyttä, haastateltavien määritelmiä ja tulkintoja ilmiöistä [HiH04]. Avoin haastattelun tavoin menetelmä mahdollistaa vapaamuotoisen ajatustenvaihdon, mutta toisaalta tuo haastatteluun rakennetta valmisteltujen teemojen muodossa.

Haastateltaviksi on tärkeää valita sellaisia henkilöitä, joilta voidaan olettaa saatavan hyödyllistä tietoa tutkimuksen kohteena olevista asioista [Sap06]. Hirsjärven ja Hurmeen mukaan haastateltavien sopivaan lukumäärään vaikuttaa tutkimuksen tarkoitus ja tutkimusongelma.

Koska laadullisen tutkimuksen tavoitteena ei ole tehdä tilastollisia yleistyskiä, vaan ymmärtää jotakin asiaa syvällisesti, rikas ja tutkimuksen kannalta riittävä aineisto saattaa muodostua jo muutaman haastattelun pohjalta [HiH04].

Tässä tutkimuksessa tehtiin yhteensä seitsemän haastattelua kuudelle henkilölle ajanjaksolla 5.–11.4.2020 (taulukko 3). Kukin henkilö haastateltiin erikseen. Haastattelut toteutettiin Google Meet -audiopuheluin ja äänitettiin haastateltavien suostumuksella. Lyhyin haastattelu kesti 63 minuuttia, pisin 97 minuuttia. Haastateltaviksi kutsuttiin tutkijan oman verkoston kautta asiantuntijoita, joilla oli vähintään kymmenen vuoden työkokemus IT-alalta. Valintakriteereinä oli lisäksi saada mukaan erilaisia toimenkuvia edustavia henkilöitä sekä sellaisia henkilöitä, joista osalla olisi työkokemusta IT-toimituspuolella toimimisesta ja osalla IT-työn tilaajapuolella toimimisesta. Kullekin haastateltavalle annettiin tunniste anonymiteetin säilyttämiseksi.

Haastateltavan tunniste aineistossa	Haastateltavan työkokemus IT-alalta	Tiivistelmä haastateltavan toimenkuvista IT-alalla	Haastattelun ajan-kohta	Haastattelun kesto
Henkilö A	25 vuotta	Toimittajan puolella, ostajan puolella, yrittäjänä; pk- ja suuryrityksissä	6.4.2020 klo 9.00 9.4. 2020 klo 15.30	63 min + 75 min
Henkilö B	17 vuotta	Toimittajan puolella; pk- ja suuryrityksissä	6.4.2020 klo 10.30	63 min
Henkilö C	19 vuotta	Toimittajan puolella, yrittäjänä, alihankkijana; pk- ja suuryrityksissä	7.4.2020 klo 12.00	86 min
Henkilö D	12 vuotta	Toimittajan puolella, ostajan puolella; pk- ja suuryrityksissä	8.4.2020 klo 14.00	81 min
Henkilö E	20 vuotta	Toimittajan puolella, ostajan puolella; pk- ja suuryrityksissä	8.4.2020 klo 17:30	79 min
Henkilö F	28 vuotta	Toimittajan puolella, ostajan puolella; pk-yrityksissä ja julkishallinnossa	9.4.2020 klo 14:00	76 min

Taulukko 3. Koonti haastateltavista ja tehdyistä teemahaastatteluista

Teemahaastattelun teoriasta ja käytännöstä kirjoittaneet Hirsjärvi ja Hurme toteavat teemahaastattelurungon muodostuvan varsinaisten valmiiden kysymysten sijaan teema-alueista, jotka pohjaavat tutkimuksen kannalta olennaisiin aiheisiin. Ajatuksena on, että tutkijan koostamat teemat toimivat eräänlaisena muistilistana, suuntaavat keskustelua ja muotoutuvat kysymyksiksi haastattelun aikana ja vuorovaikutuksessa haastateltavan kanssa [HiH04]. Teemahaastattelua suunnitellessaan tutkijan on ratkaistava, mitkä teemat hän valitsee käsiteltäviksi sekä millä tavalla hän ottaa ne esille haastateltavien kanssa, jotta haastatteluilla saadaan kartutettua mahdollisimman paljon tutkimusta hyödyttävää tietoa.

Kuvien käyttämisen on osoitettu olevan hyödyllinen tekniikka aineistonkeruun yhteydessä muun muassa käsiteltävien aiheiden ollessa monimutkaisia. Kuvat auttavat keskustelun fokusoinnissa, asioiden muistamisessa ja reflektoinnissa [UTB11]. Kuvamateriaali on sitä toimivampi, mitä paremmin haastateltavat pystyvät tulkitsemaan sitä omilla kyvyillään haastattelun aikana. Toisaalta kuvien käyttöön liittyy myös riski siitä, että ne ohjaavat ajattelua liikaa ja sitä kautta rajoittavat ja vääristävät tiedonkeruuta [UDL08].

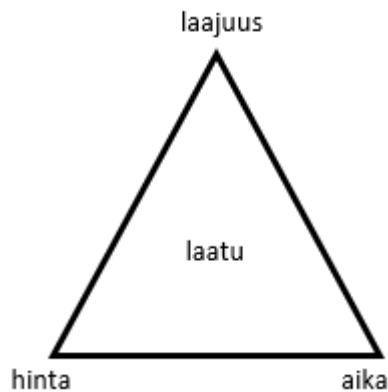
Kuvallista informaatiota voidaan käyttää eri tavoin haastattelujen tukena. Kuvat voivat esimerkiksi olla tutkijan etukäteen valmiiksi valitsemia kohteita, joita tarkastellaan haastattelussa. Kuvia voidaan myös piirtää, muokata tai tuottaa haastattelun aikana tutkijan toimesta, haastateltavan toimesta tai yhdessä [UTB11]. Törrönen käyttää termejä heräteobjekti (*stimulus object*) ja heräteteksti (*stimulus text*), jotka korostavat kuvien käyttämistä ajatusten ja keskustelun herättäjinä [Tör02]. Heräteobjekti itsessään esittää yhdenlaisen tulkinnan tutkimuksen kohteena olevasta ilmiöstä. Kun haastateltava vuorostaan esittää oman tulkintansa heräteobjektin pohjalta, jatkuva semioottinen prosessi etenee.

Heräteobjektien etsimisen voidaan ajatella olevan kuin metsästystä: tutkijat kartoittavat ja analysoivat kohdealueesta saatavilla olevaa materiaalia löytääkseen tutkimuksen kannalta hedelmällisiä heräteobjekteja. Törrönen esittää kolme eri tyypistä strategiaa, joilla heräteobjekteja voidaan käyttää haastatteluiden aikana tiedon kartuttamisessa sekä tulosten analysoinnissa. Ensinnäkin kuva voi toimia vihjeenä (*clue*), jolloin sen tarkoituksena on herättää ajatuksia ja keskustelua siitä, miten ja millaista kokonaisuutta se edustaa. Toiseksi kuva voi toimia tarkasteltavan ilmiön pienoismallina (*microcosm*), jota vasten haastateltava peilaa omia käsityksiään ja mielipiteitään. Kolmanneksi kuva voi toimia provosoijana (*provoker*), jolloin sillä voidaan esimerkiksi haastaa vallalla olevia käsityksiä [Tör02].

Tässä tutkimuksessa teemahaastattelujen teemat muodostettiin tietoperustan pohjalta etsimällä erilaisia toisiaan täydentäviä näkökulmia laatuun ja päätöksentekoon ohjelmistotuotannon kontekstissa. Tavoitteena oli päästä käsiksi tutkimuksen kannalta kiinnostaviin ilmiöihin kannustamalla haastateltavia jakamaan tilanteita, tapauksia, tarinoita ja kokemuksia, joita myöhemmin peilattaisiin tietoperustaan ja valittuihin ohjelmistotuotannon viitekehyksiin. Kunkin teeman tueksi laadittiin kuva, joka toimi teeman käsittelyn aikana haastateltavan ja haastattelijan eli tutkijan yhteisesti tarkasteltavana kohteena, ajattelun ja keskustelun vauhdittajana. Seuraavaksi esitellään teemahaastatteluihin valitut teemat ja niihin kytkeytyvät heräteobjektit eli kuvat.

3.2.1 Teema 1: Resurssit ja rajoitteet

Ensimmäinen teema käsitteli resursseja ja rajoitteita. Heräteobjektiksi valittiin vinkki- ja provosointinäkökulmasta projektinhallinnan kolmio (*project management triangle*), joka tunnetaan myös nimellä rautakolmio (*iron triangle*) tai kolmirajoitteet (*triple constraints*) (kuva 12). Kolmio on hyvin yksinkertaistettu, ja siihen liittyy huomattavia ristiriitaisuuksia ja kritiikkiä. Samalla se on kuitenkin innoittanut paljon tutkimusta ja pohdintaa käytettävissä olevien resurssien, vaatimusten ja laadun suhteesta ja vaikutuksista toisiinsa [PHA18].

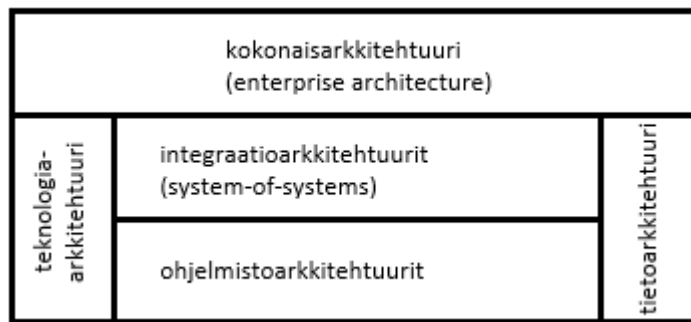


Kuva 12. Projektinhallinnan kolmio

Haastatteluissa teemaan liittyen tutkija esitti muun muassa seuraavia kysymyksiä: Onko kolmio sinulle entuudestaan tuttu? Onko laatu käytettävissä olevan rahan, käytettävissä olevan ajan ja tuotettavan laajuuden (*scope*) seurausta, niiden funktio, vai puuttuuko kuviosta jotain oleellista? Onko sinulla kokemuksia tällaisesta kolmiodraamasta?

3.2.2 Teema 2: Arkkitehtuurit

IT-arkkitehtuurit valittiin toiseksi teemaksi, sillä teoriansikin mukaan arkkitehtuureilla on selvästi vahva yhteys päätöksentekoon ja laatuun. Teeman heräteobjektiksi laadittiin TOGAF-viitekehyksessä [TOG18] määriteltyjen arkkitehtuurikerrosten pohjalta muokattu kuva. Alkuperäiseen kuvaan lisättiin integraatioarkkitehtuurien (*system-of-systems*) taso ja TOGA-Fin liiketoiminta-arkkitehtuuritaso (*business architecture*) korvattiin kokonaisarkkitehtuurilla (*enterprise architecture*) (kuva 13, s. 29). Integraatioarkkitehtuurit ja kokonaisarkkitehtuuri ovat molemmat toimialalla käytännössä usein ilmeneviä termejä, ja tutkijan kokemusperäisenä ajatuksena oli, että ne saattaisivat herättää enemmän assosiaatioita kuin TOGA-Fissa mainittu liiketoiminta-arkkitehtuuri. Kuvan pääasiallisena herätestrategiana oli toimia pienoismallina IT-arkkitehtuurien eri tasoista.

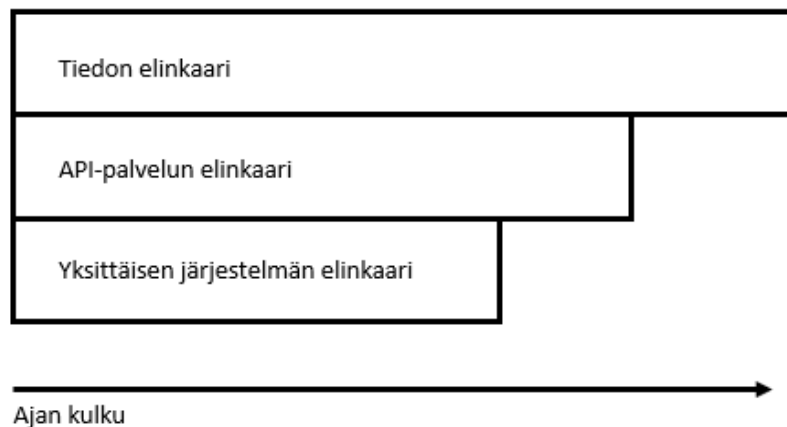


Kuva 13. Arkkitehtuurien eri tasot

Haastatteluissa tutkija esitti muun muassa seuraavia kysymyksiä: Minkä eri arkkitehtuuritasojen kanssa olet itse ollut tekemisissä? Tuleeko mieleesi kiinnostavia päätöksentekotilanteita projekteihin tai jatkuviin prosesseihin liittyen, joilla oli vaikutuksia laatuun?

3.2.3 Teema 3: Elinkaaret

Kolmannen teeman tarkoituksena oli metsästää laadun ja päätöksenteon ilmiöitä, jotka liittyvät ajan kulkuun, tapahtuvat ajan funktiona. Kuva 14 esittää eri tyyppisten entiteettien elinkaaria ja toimi vihjeenä ja keskustelun herättäjänä haastatteluissa.



Kuva 14. Eri tyyppisten entiteettien elinkaaria

Haastatteluissa kysyttyjä kysymyksiä olivat esimerkiksi: Onko tiedolla pidempi elinkaari kuin järjestelmillä? Entä integraatorajapinnoilla? Tuleeko mieleesi tapauksia, joissa ajan kuluessa jokin muutos, ehkä yllättävä, aiheutti haasteita laadun kanssa?

3.2.4 Teema 4: Toiminnalliset ja ei-toiminnalliset vaatimukset

Neljännellä teemalla pyrittiin tavoittamaan ilmiöitä, joihin jollain tavalla liittyy toiminnallisten ja ei-toiminnallisten vaatimusten suhde. Kuvan 15 tarkoituksena oli toimia heräteobjektityypiltään vihjeenä ja provosoijana. Tutkijan laatima kuva esittää kärjistetyn ajatuksen siitä, että yksityiskohtien tasolta arkkitehtuuritasolle siirryttäessä kaikki käsittelyn kannalta oleelliset vaatimukset olisivat luonteeltaan ei-toiminnallisia.



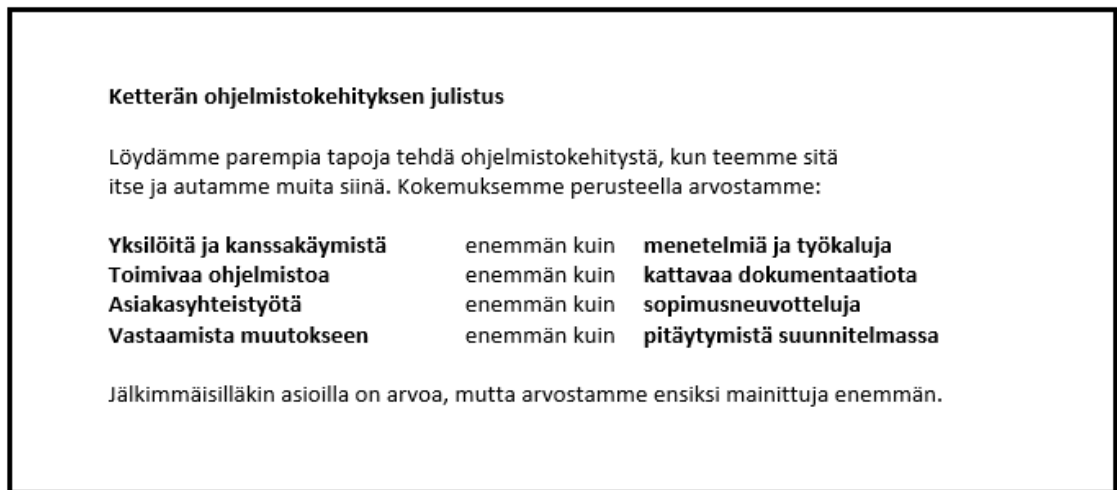
Kuva 15. Arkkitehtuurin ja vaatimusten välinen suhde

Tutkijan haastatteluissa esittämiä kysymyksiä olivat esimerkiksi: Onko niin, että siirryttäessä yksityiskohtien tasolta kohti arkkitehtuurien tasoa ei-toiminnallisten vaatimusten osuus ja merkitys kasvaa? Ovatko laatuun vaikuttavat päätökset usein sellaisia, joissa joudutaan painottamaan jotakin ei-toiminnallista piirrettä jonkin toisen piirteen kustannuksella?

3.2.5 Teema 5: Menetelmät ja lopputuotokset

Viides teema käsitteli menetelmiä ja lopputuotoksia, ja sen tarkoituksena oli tuoda laadun tarkasteluun mukaan lisää erilaisia ohjelmistotuotannon käsitteitä: dokumentaatio, menetelmät ja työkalut, projektit ja prosessit. Tässä käytettiin heräteobjektina ketterän ohjelmistokehityksen julistusta (*Agile Manifesto*) (kuva 16, s. 31). Se kuvaa asioita vastakkainasettelun kautta ja edustaa siten herätestrategialtaan eniten provosoijaa.

Tutkijan esittämiä kysymyksiä olivat muun muassa: Mitä sinusta on kattava dokumentaatio? Oletko törmännyt tilanteisiin, joissa ihmetellään esimerkiksi, miksi dokumentaatiota ei ole olemassa tai sitä ei ole pidetty ajantasaisena?

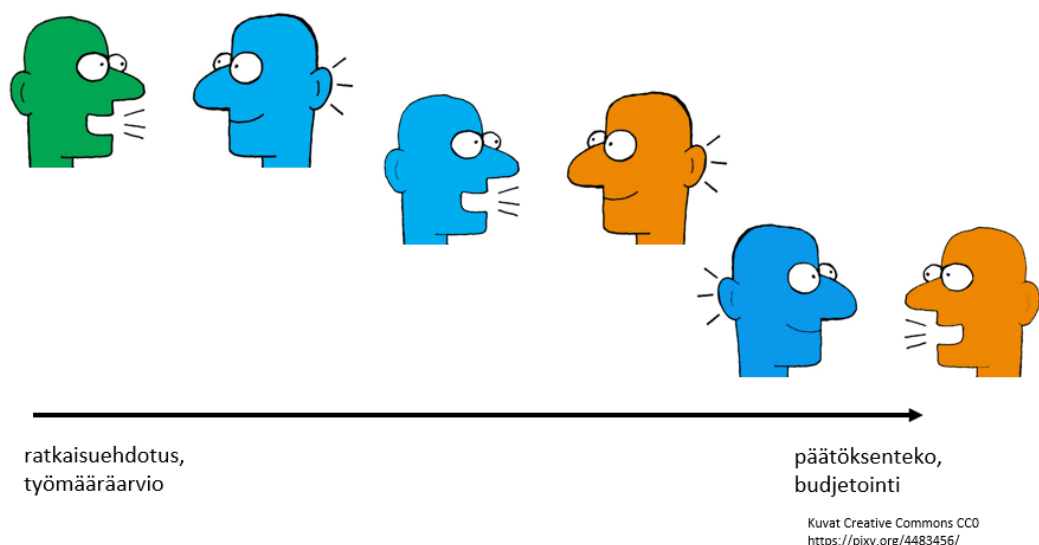


Kuva 16. Ketterän ohjelmistokehityksen julistus [BBB01]

3.2.6 Teema 6: Viestintä

Kuudes teema valittiin viestinnän näkökulmasta. Tarkoituksena oli selvittää haastateltavien kokemuksia siitä, välittykö tekninen tietämys ymmärrettävässä muodossa IT-asiantuntijoilta liiketoiminnan päätöksentekijöille. Heräteobjektina käytettiin kuvaa, jossa työmäärä-arvio kulkee työn tekijätaholta päätöksenteosta tai budjetoinnista vastaavalle taholle (kuva 17).

Haastatteluissa kysyttyjä kysymyksiä olivat esimerkiksi: Välitetäänkö kokemustesi mukaan liiketoiminnalle riittävän laajaa ja riittävän tarkkaa tietoa budjetoinnin ja päätöksenteon tueksi? Kuinka hyvin liiketoiminta on ymmärtänyt, mitä on ostamassa ja saamassa?



Kuva 17. Tiedon kulku IT:ltä liiketoiminnalle

3.2.7 Teema 7: Kognitiiviset vinoumat

Seitsemännen teeman tarkoituksena oli herättää keskustelua siitä, minkälaisia kokemuksia haastateltavilla oli kognitiivisten vinoumien vaikutuksista päätöksentekoon ja sitä kautta laatuun. Kognitiiviset vinoumat ovat rationaalisessa ajattelussa informaation käsittelyn ja tulokinnan yhteydessä tapahtuvia virheitä, jotka vaikuttavat arviointikykyyn ja päätöksentekoon [Che20]. Kun ihminen alkaa käyttää päätöksenteossa esimerkiksi intuitiotaan tai niin sanottua maalaisjärkeä selviytymiskeinona käsiteltävien asioiden ollessa liian monimutkaisia suhteissa käytettävissä olevaan aikaan tai tietoon, kognitiivisten vinoumien vaikutusten todennäköisyys kasvaa [Cap13].

Heräteobjektina toimi lista, jolle tutkija oli poiminut muutamia kognitiivisia vinoumia (kuva 18). Tutkija esitti haastatteluissa muun muassa seuraavia kysymyksiä: Oletko törmännyt tilanteisiin, joissa jokin kognitiivinen vinouma on päässyt selvästi vaikuttamaan päätöksentekoon ja sitä kautta ohjelmistotuotannon laatuun?

Kognitiivisia vinoumia	
Ankkurointivaikutus	Painotetaan liikaa ensimmäiseksi tarjottua tietoa
Vahvistusharha	Taipuvaisuus puoltaa ennakkokäsityksiä tai hypoteeseja tukevaa informaatiota
Ylivertaisuusvinouma	Itsensä yliarviointi jossakin suhteessa kuten vaikkapa jonkin taidon hallinnassa
Vahvistamisvääristymä	Yksilö panee merkille kaiken jotain oletusta tukevan ja muistaa kaiken, mikä näyttää vahvistavan sitä, mutta muistaa paljon heikommin seikat, jotka voisivat kumota oletuksen
Sädekehävaikutus	Ihmiset yleensä muodostavat mielipiteensä ihmisestä ja pitävät sen mielipiteen pitkään. Silmälasit → älykkyys. Johtajan asema → pätevyys.
Ikea-efekti, not invented here -syndrooma	Suhteettoman paljon painoarvoa annetaan jollekin jota itse tuotettu. Hyväksytään helpommin omat ideat kuin ulkopuolisten esittämät ideat.
Nirvana-virhepäätelmä	Torjutaan esitetty tai kokeiltu vaihtoehto vertaamalla sitä epärealistiseen, idealisoituun vaihtoehtoon

Kuva 18. Kognitiivisia vinoumia

3.3 Sisällönanalyysi aineiston analysointimenetelmänä

Haastatteluaineiston analysointi tehtiin käyttämällä sisällönanalyysia, laadullisen tutkimuksen perusanalysointimenetelmää. Eskolan ja Suorannan mukaan sen tavoitteena on tiivistää aineisto selkeäksi sanalliseksi kuvaukseksi, jonka perusteella voidaan tehdä johtopäätöksiä tutkittavista ilmiöistä. Kyse on prosessista, jossa aineisto pilkotaan osiin, käsitteellistetään ja kootaan jälleen yhteen uudelleenlaiseksi kokonaisuudeksi [TuS03]. Alasuutari puolestaan tunnistaa laadullisen tutkimuksen analyysissa kaksi toisiinsa kietoutuvaa vaihetta, jotka ovat havaintojen pelkistäminen ja arvoituksen ratkaiseminen [Ala11].

Sisällönanalyysin ensimmäisessä vaiheessa tutkija perehtyy kunnolla aineistoonsa ja päättää, mistä hän on siinä kiinnostunut. Käytännössä tässä redusoinniksi kutsutussa vaiheessa tutkija merkitsee aineistosta esimerkiksi alleviivaamalla oman tutkimuksensa kannalta olennaiset kohdat ja erottaa ne muusta aineistosta [TuS03]. Näin esimerkiksi runsas haastatteluaineisto saadaan pelkistettyä helpommin käsiteltävään muotoon: joukoksi tutkimuksen kysymyksenasettelun kannalta tärkeitä raakahavaintoja [Ala11]. Minkä mittaisia havainnot ovat, määrittyy tutkittavan asian ja aineiston mukaan. Analyysiyksiköksi voidaan valita esimerkiksi sana, lause tai ajatuskokonaisuus [TuS03].

Seuraavassa vaiheessa eli klusteroinnissa poimitut havainnot ryhmitellään jollakin sopivaksi havaitulla tavalla, kuten luokittelemalla, teemoittelemalla tai tyypittelemällä [TuS03]. Alasuutari kirjoittaa havaintojen yhdistämisestä, samankaltaisuuksien ja erojen hahmottamisesta analyysiyksiköiden välillä [Ala11]. Ryhmitellessään aineistoa tutkija tekee päättelyä, jossa tunnistetaan induktiivinen eli aineistolähtöinen, deduktiivinen eli teorialähtöinen sekä abduktiivinen eli teoriasidonnainen lähestymistapa.

Aineistolähtöisessä analyysissa tutkimuksen teoria rakennetaan aineiston pohjalta eikä tarkoituksena ole testata ennalta esitettyä teoriaa tai hypoteeseja. Teorialähtöisessä menetelmässä aineistoa tarkastellaan jonkin viitekehyksen, teorian tai käsitejärjestelmän kautta. Teoriasidonnainen analyysi on edellisten välimuoto, jossa teoriaa käytetään ohjaamaan aineistolähtöistä tarkastelua [EsS00, TuS03].

Sisällönanalyysin lopuksi ryhmitelty aineisto abstrahoidaan eli käsitteellistetään. Tuomen ja Sarajärven mukaan laadulliselle tutkimukselle tyypillisessä aineistolähtöisessä analyysissa tämä tarkoittaa, että jokainen ryhmä nimetään sen sisältöä parhaiten kuvaavalla tavalla. Analyysin tuloksena muodostuu joukko käsitteitä tai teoreettinen malli, jota pelitään teoriaan.

Mallin pohjalta tutkija tekee tulkintoja siitä, minkälaisia merkityksiä tutkittavat antavat asioille [TuS03]. Alasuutarin sanoin tutkija ratkaisee arvoituksen käyttämällä johtolankoina analyysissä tuotettuja havaintoja [Ala11].

Tässä tutkimuksessa teemahaastatteluilla kerätty, äänitetty aineisto litteroitiin eli kirjoitettiin auki Google Docsiin. Redusoinnissa huomio kiinnitettiin haastateltavien kertomiin tilanteisiin, tarinoihin ja kokemuksiin. Vaikka esimerkiksi Eskolan ja Suorannan mukaan teemahaastattelurunkoa voidaan yleensä pitää hyvänä lähtökohtana aineiston järjestämiselle [EsS00], tässä tapauksessa oli perusteltua toimia toisin: haastatteluihin valitut teemat olivat palvelleet ennen kaikkea keskustelun herättäjinä ja oli oletettavissa, etteivät tutkimuskysymyksen näkökulmasta kiinnostavat ilmiöt liittyisi suoraan niihin. Havaintojen ryhmittely tehtiin siten aineistolähtöisesti, ilman teemahaastattelurungon tuomaa viitekehystä.

Klusterointi- eli ryhmittelyvaiheessa haastateltavien kokemuksista alkoi hahmottua ilmiöitä ja käsitteitä, joiden pohjalta jatkettiin aineiston jäsentämistä. Aineistosta poimittiin erilleen tarkempia ilmauksia, ajatuskokonaisuuksia, joiden välillä havaittiin yhdistäviä ja toisaalta erottavia tekijöitä. Ryhmittelymenetelmänä käytettiin teemoittelua, jossa korostuu, mitä haastateltavat kertovat kustakin aiheesta [TuS03]. Oleellisena ei esimerkiksi pidetty niinkään sitä, kuinka monen haastateltavan kertomuksissa esiintyi tietty ilmiö, vaan sitä, millaisia ilmiöitä ylipäättään nousi esille ja minkä asiayhteyden kautta haastateltava itse hahmotti ilmiötä. Käytännössä Google Docsiin kirjattiin teemoja, ja niiden jokaisen alle sijoitettiin ne aineistosta poimitut ajatuskokonaisuudet, joissa käsiteltiin kyseistä teemaa.

Abstrahointivaiheessa teemojen pohjalta muodostettiin mahdollisuuksien mukaan niitä tarkemmin kuvaavia käsitteitä, joille haettiin vahvistusta tietoperustasta ja muista tutkimusjulkaisuista. Tarkoituksena oli selvittää, oliko kunkin teeman yhteydessä käsitellyille asioille olemassa toimialalla vakiintunutta käsitettä ja mikä tuo käsite oli, esimerkiksi käytetäänkö yleisesti käsitettä tekninen laina vai tekninen velka. Sisällönanalyysin tuloksena tunnistettiin ohjelmistotuotannon päätöksentekotilanteissa joukko laatua haastavia ilmiöitä, jotka käsitteellistettiin seuraavasti: niukkuus, tiedon määrä ja laatu, tekninen velka, ulkoisen ympäristön muutos, tietojärjestelmän kriittisyys, tulevaisuudenkestävyys, omistajuus ja proaktiivisuus sekä kognitiiviset vinoumat.

4 Tulokset

4.1 Laatua haastavat ilmiöt

Seuraavissa alaluvuissa käydään läpi haastatteluissa esiin nousseet ilmiöt. Ilmiöitä esitellään suorien haastattelulainauksen avulla ja peilaten niitä tietoperustaa ja muuta kirjallisuutta vasten. Kunkin ilmiön kohdalla tuodaan esille, miten se liittyy laatuun ja päätöksentekoon.

4.1.1 Niukkuus

Haastatteluissa mainittiin neljä talouteen liittyvää käsitettä (taulukko 4) joista jokaisella voi olla merkittävä laatuun vaikuttava rooli päätöksenteossa. Yhteiseksi otsikoksi valittiin niukkuus, joka tunnistettiin keskeiseksi taustalla olevaksi tekijäksi.

Käsitteen nimi	Kuvaus
Niukkuus (<i>scarcity</i> , <i>paucity</i>)	Taloustieteen perusongelma, jolla tarkoitetaan erotusta käytettävissä olevien rajallisten resurssien ja teoreettisesti rajattoman resurssintarpeen välillä. Siitä aiheutuu tarve tehdä päätöksiä resurssien allokoinniseksi tehokkaasti [Inv20].
Kustannus-hyötyanalyysi (<i>cost-benefit analysis</i>)	Arvioitujen hyötyjen ja arvioitujen kustannusten erotus. Myös vaihtoehtoiskustannukset tärkeä huomioida mukaan. Laadukas analyysi voi tapauksesta riippuen olla työmäärältään ja haastavuudeltaan merkittävä tutkimus itsessään [Inv20]
Vaihtoehtoiskustannus (<i>opportunity cost</i>)	Potentiaalinen hyöty joka menetetään kun tietty vaihtoehto jätetään valitsematta [Inv20].
Riskianalyysi (<i>risk analysis</i>)	Prosessi jossa tunnistetaan mahdollisia negatiivisia tapahtumia, niiden todennäköisyyksiä ja vaikutuksia [Inv20].

Taulukko 4. Esille nousseet talouden käsitteet

Resurssien niukkuudessa (*scarcity*, *paucity*) on kyse taloustieteen perusongelmasta, jolla tarkoitetaan eroa tarjolla olevien rajallisten resurssien ja teoreettisesti rajattoman resurssintarpeen välillä. Siitä aiheutuu tarve tehdä päätöksiä resurssien allokoinniseksi tehokkaasti [Inv20]. Laadun puolestapuhujat pyrkivät rakentamaan eetosta, joka kannustaa päätöksentekotilanteissa investoimaan laatuun sillä perustelulla, että hyvä laatu maksaa itsensä takaisin [JuG98, Ste95]. On kuitenkin tilanteita, joissa niukkuus on niin ankara, ettei rahaa yksinkertaisesti ole käytettävissä ja joudutaan tekemään kompromisseja, kuten joustamaan laadusta:

“olen ollut useammassakin – asiakkuudessa, joissa nimenomaan kohtuulliset kustannukset on ollut se draiveri – asiakas on esimerkiksi äärimmäisen pieni ja niiden liiketoiminta saattaa pyöriäkin kokonaan tuollaisen ratkaisun päällä, joka on tehty matalilla kustannuksilla ja se ei ole laadullisesti kovin hyvä. Objektiivisesti tarkasteltuna se on silloin liiketoimintakriittinen, mutta käytännössä usein sellaisella tavalla, joka ei vaikuta päätöksentekoon, koska asiakkaalla ei ole kykyä priorisoida sen liiketoimintakriittisyyden mukaan, koska niillä on esimerkiksi rahat vähissä koko ajan.” (henkilö C)

Haastatteluissa tuotiin esille myös tilanteita, joissa niukkuus on liittynyt henkilöresursseihin. Rajoittavana tekijänä niissä on ollut henkilötyövuosina mitattava efektiivinen henkilöstömäärä, yksittäisille henkilöille tai rooleille keskittyvä erityisosaaminen tai organisaation hallinnon ja johtamisen vastuunjaon kautta annettu päätöksenteon mandaatti:

“On pieni tiimi – [jotakin] henkilötyövuotta tällä hetkellä. Tämän mittaisessa organisaatiossa ei tietysti sillä määrällä kaikkea pysty tekemään.” (henkilö F)

“Jos arkkitehtuurityö 'hankaloittaa' projektin toimintaa, projektissa on painetta tehdä nopeammin asioita – oikotiet helpompia.” (henkilö B)

“Päättäjällä ei välttämättä ole aikaa – jos vaatii työpanosta juuri häneltä, että jotain asiaa aletaan kilpailuttamaan tai tekemään. Se voi olla juuri se kynnyks, jos on muitakin [tekemistä] paljon, sitoo tiettyjen henkilöiden työpanosta.” (henkilö F)

Dickinson et al. käsittelevät niukkuuteen liittyvää priorisointiongelmaa johtamisen (*management*) ja johtajuuden (*leadership*) näkökulmasta terveydenhoitoalalla, missä IT-alan tavoin systeemit ovat monimutkaisia ja muuttuvat nopeasti [DFR11]. He käyttävät Grintin [Gri05] luokittelua ongelmille (kriisit, kesyt, viheliäiset) ja ehdottavat, että eri tyyppisiin tilanteisiin sovelletaan eri tyyppistä johtamista tai johtajuutta. Kriisitilanteet vaativat tiukkaa komentoa ja mahdollisesti pakottamista, kesyt ongelmat 'tekniistä', suorittavaa johtamista ja viheliäiset ongelmat tilanteeseen sopeutuvampaa johtajuutta ja politiikan tajua, joka voi hyötyä esimerkiksi merkityksellistämisestä [Wei95].

Kustannus-hyötyanalyysi, vaihtoehtoiskustannus ja riskianalyysi nousivat esille keskusteltaessa siitä, saavatko päätöksentekijät riittävästi ja riittävän oikeellista, kattavaa ja laadukasta informaatiota päätöksenteon tueksi.

“Päätöksenteko lähtee aina liiketoiminta-arvosta. Pääsääntöisesti kolmea kategoriaa: Mitä euroina tällä ko. projektilla voitaisiin saada, mikä sen pointti on, haluaako joku maksaa siitä. Toinen on se että, no jos ei kukaan halua siitä maksaa, niin korjaako se jotain ongelmaa, mistä me saadaan [muuta] asiakasarvoa – kolmas on ns. putkiremppoja: näitä ei kukaan haluaisi tehdä ja laittaa rahaa, mutta joskus on vain se [tieto]kanta päivitettävä tai haettava Verkkokaupasta uudet serverit” (henkilö E)

“Ehkä toinen puoli on sitten riskianalyysi: mitkä riskit liittyy siihen, että työ jätetään tekemättä vs. ne riskit että [työ] tehdään. Aika monesti tällaisissa vähänkin isommissa päätöksissä on siis molempiin suuntiin riskejä ja niitä pitää kyetä jotenkin balansoimaan – opportunity cost: mitä mahdollisuuksia menetetään jos ei tehdä tätä.” (henkilö C)

Haastatteluissa todetaan, että kustannusten suuruusluokkien hahmotus voi olla kuitenkin vaikeaa sekä asiakkaille että toimittajille. Kustannus-hyötyanalyysiä monimutkaistaa lisäksi se, että tavoiteltavat hyödyt saattavat olla epäsuoria:

“harvat asiakkaat on hyviä ostamaan – IT-alan lainalaisuudet eivät ole heille selkeitä – mikä on helppoa, mikä on vaikeaa. Jos et hahmota asioiden suuruusluokkaa niin sehän toki vaikeuttaa hyvää päätöksentekoa cost-benefit -tyyppisessä analyysissä.” (henkilö C)

“Toimittajalla usein unohtuu, että pitäisi olla nopeasti käsitys, että mitä tämän pitäisi maksaa. Että onko kymmentuhatta euroa paljon vai vähän, vai onko tämä sadan tonnin vai puolen miljoonan keikka. Täsmälleen samalla tarjouspyyntöaineistolla tarjoukset saattaa heittää, että toinen tarjoaa kolmella-kymmenelläviidellä tonnilla ja toinen tarjoaa neljäsataaviiskyt tonnia.” (henkilö E)

“[Päätöksentekijälle tärkeitä ovat] huomattavan suora rahallinen säästö, tai sitten sen tason tekninen parannus, joka mahdollistaa esimerkiksi kommunikoinnin uusien järjestelmien kanssa, joka sitten lopujen lopuksi tuo rahallista säästöä tai jotain prosessien nopeutumista.” (henkilö D)

Kustannus-hyötyanalyysia on kritisoitu myös muun muassa siitä, että se edellyttää arvioitavien hyötyjen ja haittojen yhteismitallistamista rahassa, mikä ei ole aina mahdollista tai eettisesti oikein [Fra00]. Lisäksi kustannus-hyötyanalyysi tarvitsee tuekseen riittävän rationaalisen päätöksentekoprosessin. Vaikka organisatorisessa päätöksenteossa pyritään rationaalisuuteen, täydelliseen rationaalisuuteen tuskin koskaan päästään sen ankarien kriteerien johdosta [Hyy12].

4.1.2 Tiedon määrä ja laatu

Päätöksenteossa käytettävissä olevan tiedon määrän ja laadun merkitys nousi haastatteluissa esille keskusteltaessa työn tilaajan (asiakas) ja työn toteuttajan (myyjä, toimittaja) välisestä kanssakäymisestä. Palvelun tai tuotteen ostaminen on tilanne, jossa myyjällä ja ostajalla on tarve toimivaan yhteistyöhön, mutta koska kumpikin ajattelee myös omaa etuaan, mukana on aina intressien ristiriita.

“Myyjät lupaavat yleensä paljon. Paljon ollaan myyjien kuvausten varassa.” (henkilö F)

“Pari klassista ongelmaa edelleen: se mitä tarvitaan ja se mitä tarjotaan – ei edelleenkään kohtaa – ei oikein keretä keskittyä – tarjotaan vähän sinnepäin jotain, jotta se diili näyttää houkuttelevalta. Se lopulta ei ookaan ihan sitä, mitä tarvittiin tai ei kata kuitenkaan koko hommaa. Asiakkaan puolella on edelleen haasteita ostaa. Helposti ostetaan valmiiksi [teknistä] velkaa – [Myyjän pitäisi] ymmärtää asiakkaan ongelma, eikä tarjota vaan Drupalia, koska meillä nyt sattuu olemaan kaksi Drupal-jätkää penkillä.” (henkilö E)

”Saattaa johtaa – ristiriitatilanteisiin, jos asiakas kokisi, että [hankinnasta] saatava hyöty olisi huomattava, mutta he uskovat, että se pitäisi saada matalammalla kustannuksella. Ja me taas toimittajana todetaan sitten, että jos tämä tehdään matalilla kustannuksilla, niin sitten siitä ei tule taas sitä laadusta jälkeä.” (henkilö C)

Tilannetta hankaloittaa ohjelmistotuotannon monimutkaisuus. Erityisesti laadulliset ominaisuudet, kuten robustius ja suorituskky, voivat olla vaikea alue molemmille osapuolille:

”Ostotilanteessahan käytännössä yritän ostaa jotain asiaa, jota en tunne. Joku on myymässä asiaa, jota hänkään ei tunne kovin hyvin – itse tekijähän on kauheen harvoin mukana [selittämässä] sitä asiaa. Jolloin – joku lupaa jotain, ja mä luulen että se tarkoittaa jotain. Sitten kun me projektia tehdään, niin käykin sen tekijän toimesta ilmi, että ei tää nyt olekaan ihan niinku ollaan puhuttu. Esim. tietyn ison toimittajan verkkokaupan optimointi: Verkkokauppa itsessään maksaa tietyn summan, mutta sen optimointiin saa laittaa toisen mokoman. Eli sen voi optimoida tosi hyvin, mutta se on ääri vaikeaa. Tämän tyyppinen kustannus ei tule helposti ilmi hankintavaiheessa. Pelkät suorituskykytestit maksaa kymppitonnin. Laaja ja monimutkainen platformi, mikä on toki hyvä, mutta sen robustiksi ja nopeaksi tekeminen on niinkuin äärettömän iso ponnistus.” (henkilö E)

Viestinnän tarve korostuu tilanteissa, joissa päätöksentekoon osallistuvilla tahoilla on erilaiset osaamisalat:

“Teknologisen ymmärtämyksen ja bisnesymmärtämyksen välillä on tosi iso gäppi vielä – tokkopa sitä koskaan tullaan kokonaan kursimaan kokonaan kasaan. Teknologiset ihmiset eivät välttämättä osaa ymmärtää bisnestä, ja ne, jotka on hakoja bisneksessä, ei välttämättä ole teknisiä – Mun mielestä on ollut tosi selkeästi nähtävissä melkein kaikissa asiakkaissa, jotka tekee digiloikkaa nyt, että bisnes asettaa entistä enemmän tosi konkreettisia vaatimuksia IT:lle, ja IT:llä on enemmän sananvaltaa bisnekselle, että miten ne tekee asioita. Ne liittyy toisiinsa niin vahvasti nykyään, bisnes ei ole enää erillään järjestelmistä. Kyllä tämä vastuuttaa meitä teknisiä osaajiaakin enemmän. – Ei lähdetä purkamaan sitä teknologiaa, ei lähdetä puhumaan RESTistä ja verbeistä, http:stä, vaan puhutaan käsitteistä: API-management ja -rajapinnat, minkä takia ne on tärkeitä, [ja niin edelleen]. Ehkä pointtina on löytää se yhteinen kieli.” (henkilö D)

Jos päätöksentekijällä eli ostajalla on riittävää teknistä ymmärrystä, hän voi pystyä optimoimaan hankintoja esimerkiksi joustamalla mahdollisuuksien mukaan jonkin toiminnon laadussa:

“Sanotaan vaikka integraatio taustajärjestelmään: vie tilaukset kerran kahdessa tunnissa ja täyttää [inventaarion] kerran yössä – kestää kuusi tuntia – mutta ei sillä ole oikeastaan mitään merkitystä, koska se ei vaikuta mihinkään. Semmoista ei tarvii turhaan lähteä optimoimaan ja käyttää siihen rahaa. Joskus se hitaus tai heikko laatu ei vaikuta siihen kokonaisuuteen, joskus se on taas oleellinen – Jos yö ei enää riitä siihen siirtoon, niin sittenhän se on pakko vaan optimoida sen verran että yksi yö riittää.” (henkilö E)

Puutteellisen tai epäonnistuneen viestinnän aiheuttamia laatuongelmia voi ilmetä, vaikka molemmat osapuolet ovat perillä samoista IT-maailman teknisistä käsitteistä eikä varsinaista selittävää tulkkausta tarvita:

“Useassa organisaatiossa on erillinen toimittaja tai tiimi, joka hoitaa integraatioita, kuin se, joka tekee järjestelmätoimituksia, projektitoimittaja. – Kaksiteräinen miekka. Integraatiotiimi saattaa ymmärtää järjestelmät ja datat paremmin, kuin jonkin projektin toimittava osapuoli, koska ovat toimineet organisaation EA:n [kokonaisarkkitehtuurin] parissa pitempään. Mutta vaatii hyvää kommunikaatiota. Tieto pitää saada kulkemaan riittävän hyvin eri toimijoiden välillä, jotta vältetään järkyttävä iteraatioitten määrä.” (henkilö B)

Yhteenvetona haastateltavat toivat esille useita kokemuksia tilanteista, joissa päätöksentekossa käytettävissä oleva tiedon määrä ja laatu ovat vaikuttaneet päätöksentekoon kokonaisuutena. Tällaisissa tilanteissa kyky viestiä monimutkaisia teknisiä asioita ja tulkata niitä ymmärrettävään muotoon korostuu.

4.1.3 Tekninen velka

Avgerioun et al. määritelmää vapaasti suomentaen tekninen velka on kokoelma ohjelmiston suunnittelu- tai toteutustason rakenteita, jotka ovat tuottaneet jotakin lyhyen tähtäimen hyötyä mutta jotka saattavat hankaloittaa myöhempää kehitystyötä tai tehdä sen jopa mahdottomaksi. Teknisen velan vaikutus on usein nähtävissä ohjelmiston sisäisten laatupiirteiden kautta, kuten ylläpidettävyydessä ja kehitettävyydessä [AKO16].

Teknisen velan käsite on aktiivisen tutkimuksen kohteena muun muassa meneillään olevan, vuonna 2017 aloitetun InsightTD-tutkimushankkeen kautta [INS20]. Siihen liittyvä Riosin, Neton ja Spínolan sateenvarjokatsaus [RMS18] laajentaa Avgerioun määritelmää tunnistamalla viisitoista erilaista teknisen velan alaluokkaa, esimerkiksi ihmiset ja dokumentaatio. Teknisen velan ilmentymistä on tutkittu muun muassa tietokannan rakenteissa [AIB18], ohjelmistoarkkitehtuurien tasolla [NOK12], serverless-arkkitehtuureissa [LDM20], yrityksen kokonaisarkkitehtuuritasolla [HHS19] sekä yksittäisen ohjelmiston komponenteissa lähdekooditasolla [ZWI13]. Teknisen velan ottaminen voi tapahtua hyvin tietoisesti ja sopimalla siitä huolellisesti päätöksenteosta vastuussa olevien tahojen kanssa:

“Projekti saattaa olla muuten hyvässä kuosissa, mutta joltain tietyltä osalta löytyy teknistä velkaa – – [mikä] johtaa siihen, että tulevat työt on kalliimpia ja hitaampia kuin niiden tarviisi olla, tai huonolaa-tuisempia. Toimittajanäkökulmasta näinä päivinä me saatetaan tehdä esimerkiksi ihan tiedostaen ja asiakkaan kanssa avoimesti kommunikoiden siten, että [sanotaan asiakkaalle] ‘me voidaan tehdä tämä tällaisella tavalla, joka sopii esimerkiksi teidän tämän vuoden budjettiin paremmin, mutta olkaa tietoisia siitä niin että heti kun tässä asiassa halutaan mennä pidemmälle, niin täytyy maksaa se tekni-nen velka takaisin, eliikkä sitten se seuraava vaihe kokonaisuudesta ei tule esimerkiksi menemään yhtä nopeasti.’” (henkilö C)

Kuten edellisessä luvussa kuvatuissa tapauksissa, myös teknisen velan tietoiseen ja hallit-tuun ottamiseen liittyy se tarve, että päätöksentekijällä on riittävän paljon riittävän laadu-kasta tietoa tukena. Tiedon puutetta voi korvata luottamuksellinen suhde ulkoiseen toimi-jaan, jos heiltä saatu tieto vaikuttaa riittävän uskottavalta:

“Meillä on sekä sellaisia asiakkaita jotka, kun heille perustelee että tästä syntyy sellaista ja sellaista velkaa, niin he saattaa hyvinkin valita sen, että he maksaa enemmän siitä [välttääkseen velan otta-mista]. Se tietysti edellyttää tosi voimakkaasti sitä, että joko asiakas itse ymmärtää sen teknisen ym-päristön riittävän hyvin, että he kykenee itse arvioimaan ja toteamaan, että se pitää paikkansa. Tai vaihtoehtoisesti luottamussuhde on sillä lailla kunnossa, että he eivät epäile, että heiltä kiskotaan yli-määräistä rahaa tarpeettomasti. On nähty myös sitä, että teknisen velan ottamisella on ostettu kalente-riaikaa – mikä voi olla hyvinkin viisasta opportunity cost -näkökulmasta” (henkilö C)

Haastateltavilla on kokemuksia sekä tilanteista, joissa teknistä velkaa maksetaan myöhem-min pois, että tilanteista, joissa velan kerryttämistä velan päälle jatketaan:

“Vaikka ehkä päätöksentekovaiheessa tunnistetaan, että otetaan teknistä velkaa, tavallaan se unohtuu, ja sit se muodostuu semmoiseksi niinkuin muuttumattomaksi tosiasiaksi, että meillä nyt on tällainen vanha järjestelmä täällä mitä ei voi vaihtaa. Enemmän ehkä keksitään tekosyitä miksei siitä [velasta] päästä eroon, kuin että tunnistetaan että se on velkaa, joka pitää jossain vaiheessa maksaa takaisin päivittämällä se järjestelmä.” (henkilö D)

”Aika monesti nykyään sitä ihan tiedostaen maksetaan takaisin, ei ehkä niin nopeasti kuin se olisi järkevää, mutta kuitenkin. Ja sitten on tietysti tahoja, joiden kanssa sitä teknistä lainaa otetaan, kunnes ollaan siinä pisteessä, että siitä on vaikea jatkaa eteenpäin.” (henkilö C)

“Kyllä tällaisia tilanteita on, että joudutaan tekemään kompromissi, että jätetään joku vanha palvelin käyttöön. [Ajetellaan, että] sieltä haetaan tietoa ja uusitaan sitten joskus, mutta ei kuitenkaan koskaan uusita. – Tietohallinnossa näitä hallitaan tiketointijärjestelmillä ja viikkopalaverilla. Johtoa ei välttämättä kiinnosta se, onko teknistä velkaa, kunhan homma toimii, niin se on ok. Kyllähän se varmaan osittain on, että mielellään tehdään se minimi – oikaistaan, ja myöhemmin joudutaan tekemään sitten kokonaan uudestaan, kun sen olisi melkein samalla rahalla tehnyt heti.” (henkilö F)

Vastaan tulee myös tilanteita, joissa ei onnistuttu ennustamaan tulevaisuutta ja hankinta osoittautuu epäonnistuneeksi, minkä seurauksena kokonaisen tietojärjestelmän voidaan ajatella muuttuvan yhdenlaiseksi teknisen velan muodoksi:

“Aina tarkoitus on projektien aikana maksaa tekninen laina tietyn scopen osalta takaisin. Mutta se suurin ikävä puoli, mikä tulee aina silloin tällöin – [otetaan] tyypillinen esimerkki: eräässä organisaatiossa kilpailutuksen jälkeen ostettiin tietty ison toimittajan verkkokauppa-alusta. Projekti ei ehtinyt edes maaliin, niin tuotteen toimittaja ilmoitti alustan tuen loppumisesta. Tyypillinen case miten menee todella ‘vihkoon’ mitä kukaan ei olisi voinut arvata. Käytännössä ostin kalliilla ison lainan. Harvinaista, mutta kaikista raskainta. Riskiarvio meni käytännössä teknologiastackin riskin vähentämiseen, mutta ei taas sitten osattu arvella sitä, että koska tuotteella ei ollut maailmalla riittävästi nostetta, niin toimittaja ei siksi tullut sitä jatkamaan.” (henkilö E)

Haastatteluissa nousi esille myös tilanne, jossa organisaatio joutui ottamaan teknistä velkaa itsestään riippumattomista syistä, ulkopuolisten tekijöiden takia. Henkilö E kertoo järjestelmäintegraatioprojektista, jossa jouduttiin rakentamaan integraatio toisen yrityksen tietojärjestelmään arkkitehtuurilla, mihin liittyi huomattavia laatuongelmia:

“Se [ulkoisen toimijan tuotteen käyttöliittymä] toimi – virtuaalikonsolin kautta – desktopilla. Ei ollut mitään automaatiota eikä mitään. Silloin oli jo iso tarve automaatiolle – Se johti siihen, että meillä oli pöytäkoneita toimiston nurkassa, jotka pyöritti makroja, joilla tehtiin automaatiokriptit. Joku oli tehnyt skriptit, ja nauhoituksella otettiin hiiren liikkeitä talteen: mihin kohtaan hiiri laitetaan ja painetaan nappeja. Olen naureskellut, että RPA:han [robotic process automation] oli sitä, mitä tehtiin jo silloin – Se on ehkä vähän älykkäämpää nykyään, koska – jos ikkuna saattoi avautua väärään kohtaan ruudulla, niin prosessi meni täysin rikki. Toi on ehkä klassinen ja räikeä esimerkki siitä, että kun ei ollut korvaavaa järjestelmää, niin jouduttiin tekemään hirveitä purkkaratkaisuja, jotka tavallaan kerrytti teknistä velkaa organisaation omaan IT-ympäristöön.” (henkilö E)

Tässä tapauksessa teknisen velan ottamisen juurisyyt tulivat organisaation ulkopuolelta, arvoketjussa olevalta kumppanilta, jonka palvelusta organisaatio oli riippuvainen. Kyseinen yritys teki tuona vuonna voittoa yli 400 miljoonaa euroa, joten järkevien rajapintojen puutteen juurisyyn on vaikea kuvitella olevan ainakaan rahallisten resurssien niukkuus.

McGregor et al. ovat tutkineet vastaavaa ilmiötä, jossa useamman eri organisaation muodostamassa arvoverkostossa yhden toimijan ottama tekninen velka näkyy ja vaikuttaa toisiin

toimijoihin. Heidän ajatuksenaan on ollut tuottaa strategisen päätöksenteon tueksi tietoa, joka auttaa muun muassa päättämään, mitä kumppaneita kannattaa valita tai mihin konsortioihin hakeutua [MMZ12]. Ohjelmistotuotantopalveluita ulkoisilta konsulttiyrityksiltä hankkiva henkilö E antaa esimerkin tapauksesta, joka edustaa teknisen velan sosiaalista alaluokkaa:

“On aina, järjestäen, joka ikinen kerta paljon kalliimpaa vaihtaa – toimittaja ja opettaa [uudet] henkilöt uudelleen siihen hommaan, vaikka ne tulee kuinka suurella innolla. Se on aina kaikkein kallein vaihtoehto, sitä ei tee pienin perustein.” (henkilö E)

Jones ja Bonsignour mainitsevat tutkimuksista, joiden mukaan noin 50 % ylläpitotyötä tekevän ohjelmoijan ajasta kuluu siihen, että hän yrittää ymmärtää, miten olemassaoleva sovellus toimii, ennen kuin yrittää tehdä siihen muutoksia [JoB12].

Fowler esittää nelikentän avulla kaksi akselia [Fow09]. Yhtenä akselina on tietoisuuden aste (*inadvertent – deliberate*): teknistä velkaa syntyy tietoisien päätöksenteon tuloksena tai epähuomiossa. Toisena akselina on harkitsevaisuuden tai huolellisuuden aste (*reckless – prudent*). Huolellisesti suunnitellulla ja sidosryhmien kanssa sovitulla lainan ottamisella, kohdentamisella, rajaamisella ja takaisinmaksulla saatetaan pystyä minimoimaan teknisen velan negatiivisia vaikutuksia ja kustannuksia sekä maksimoimaan sen hyödyt ja sitä kautta saatava oppi.

Yksi näkökanta on, että huolimattomasti ja suunnittelematta syntynyttä ”sotkua” ei pidä ilmaista teknisenä velkana [UnB09]. Tekemällä selkeä pesäero hallitsemattomasti aiheutetun sotkun ja teknisen velan välillä voidaan viestinnällisesti korostaa jälkimmäisen olevan luonnollinen, hyväksyttävä ja tarpeellinenkin osa onnistumiseen tähtäävää ohjelmistotuotantoa. Tekninen velka metaforana voi auttaa monimutkaisten taustalla olevien teknisten haasteiden ja kompromissitilanteiden viestimistä ja päätöksentekoa ihmisten kanssa, joilla ei ole IT-alan teknistä kokemusta [Fow09].

Yhteenvedona tekninen velka tunnistettiin haastatteluissa yhdeksi eniten koetuista ja merkittävimmistä laadun haastajista ohjelmistotuotannon päätöksenteossa. Suuressa osassa haastateltavien kokemista tapauksista tekninen velka on ollut tietoisien päätöksenteon kautta syntynyttä ja harkittua. Taustalla olevat syyt ovat liiketoiminnasta lähteviä. Esille nousseita perusteita ovat olleet käytettävissä olevan rahan tai ajan niukkuus tai tarve jostain muista taloussuunnittelun syistä jakaa kustannuksia seuraavan budjettikauden puolelle. Laajemmassa organisaatioiden arvoverkostossa ulkoisten toimijoiden kautta voi tulla rajoitteita, joihin ei ole pyritty tai onnistuttu vaikuttamaan. Haastateltavat toivat esille myös kokemuksia tiimien

vaihtumiseen liittyvästä sosiaalisen velan muodosta Se, missä määrin teknisen velan ottoa, syitä ja vaikutuksia on viestitty rahoituksesta ja päätöksenteosta vastaaville liiketoimintatahoille, vaihtelee runsaasti. Joissain tapauksissa tällaista ei ole viestitty lainkaan, jolloin ei ole tietoa siitä, missä määrin IT-johto ja liiketoiminta tiedostavat teknisen velan olemassaolon ja luonteen. Toisissa tapauksissa viestintä on aikaansaanut muutoksia päätöksentekoon.

4.1.4 Ulkoisen ympäristön muutos

Henkilö A nostaa esille esimerkin tilanteesta, joka liittyy toimialan suureen muutosnopeuteen ja siitä aiheutuvaan haasteeseen saada pidettyä yllä henkilöstön riittävää osaamistasoa, jotta ohjelmistotuotannossa voidaan tuottaa haluttua laatua:

“Onko sovellustoimittajalla kykyä omissa tiimeissä uudempaan tekniikkaan, vai tarvitaanko joku uusi ihminen taloon sisälle opettamaan ‘hei nyt me koodataankin tällä tavalla asioita, tämä on nyt se tämän hetken trendi.’? – Tahti, jolla ihmisen pitää pystyä omaksumaan uusia asioita, voi vaihdella runsaastikin ekosysteemistä riippuen ja olla iso haaste.” (henkilö A)

Tämä on esimerkki ajan funktiona tapahtuvasta ilmiöstä, jolla on jonkinlainen vaikutus ohjelmistotuotantoprosessiin tai sen tuotoksiin. Tutkija itse nostaa esille kaksi esimerkkiä, jotka liittyvät tietoturvaan. Ensimmäisessä tapauksessa sisäverkossa ollut hajautettu järjestelmä käytti muiden järjestelmien kanssa keskustellessaan salaamattomia protokollia (ftp, http). Päätös salaamattomien protokollien valinnasta oli tehty kustannusten säästämiseksi. Myöhemmin vuosina asenne tietoturvaa kohtaan alkoi maailmanlaajuisesti muuttua, mikä näkyi muun muassa zero trust -nimellä kulkevan ajattelumallin [Ash19] yleistymisenä. Sen valossa aiemmin tehdyt päätökset ja niiden seurauksena tehdyt ratkaisut eivät enää täyttäneet tietoturvan laatukriteereitä. Tietojärjestelmän laatu oli tästä näkökulmasta muuttunut, vaikka järjestelmään itseensä ei oltu tehty mitään muutoksia.

Toisessa tapauksessa organisaation eräs lähitulevaisuudessa korvautumassa oleva tietojärjestelmä (vanha järjestelmä) käytti ajoalustan versiota, joka oli jo poikkeuksellisen vanha suhteessa kaikkiin muihin organisaation järjestelmiin. Kustannusten säästämiseksi portausta uudempaan ei haluttu tehdä. Järjestelmä kutsui ulkoisia palveluita https-protokollalla yhteiskäyttöisen kuormantasaimen kautta. Kun kuormantasaimella yritettiin disabloida vanhentuneet TLS-protokollaversiot, muutos ei aiheuttanut ongelmia organisaation muiden tietojärjestelmien kohdalla. Vanhaan järjestelmään jouduttiin kuitenkin tekemään varsin aikaa-vievä operaatio, jotta se saatiin tukemaan uudempia TLS-versioita. Muutosten tekeminen tuli yllätyksenä ja yksi lisähaaste ja hidaste oli saada perusteltua muutos ja hankittua sille

rahoitus. Tämän prosessin kestäessä kuormantasaimen tietoturvapäivityksen kanssa jouduttiin odottamaan. Yhden järjestelmän kohdalla säästäminen vaaransi useiden järjestelmien tietoturvan pitkäksi aikaa.

Henkilö C antaa esimerkin ilmiöstä, jossa jonkin arvoverkostosta hankitun palvelun kustannusrakenne muuttuu radikaalisti:

“SSL on hyvä esimerkki siitä, että sen käyttöönottoa piti aikaisemmin perustella siksi, että [se lisää] tiettyjä vuosikustannuksia ja – pitää hallinnoida sertifikaatteja ja huolehtia niiden uusimisesta. Sitten tulee joku tällanen asia kuin Let’s Encrypt – ’tässä on sulle työkalut, millä sä automatisoit sen, ja tää ei maksa mitään, anna palaa’, se muuttaa koko ekonomian sen ympärillä. Onko termi disruptor osuva? Se on eräänlainen. Tässä tapauksessahan se ei disruptoinut juuri muuta kuin kaupallisten CA:iden bisnesmallia, mutta kyllä se IT-teknisestä näkökulmasta muutti aika paljonkin.” (henkilö C)

Ilmiö tunnistetaan myös alan kirjallisuudessa ja tutkimuksessa. Jones ja Bonsignour kuvailevat ohjelmiston laadun kontekstuaalista luonnetta perustavaa laatua olevaksi haasteeksi: Tietyissä tilanteissa elegantiksi koettu ratkaisu voi olla toisessa tilanteessa täysin toimimaton. Vaihtoehtoisesti tietyissä olosuhteissa korkeasti luotettava järjestelmä saattaa kaatua nopeasti toisenlaisten olosuhteiden vallitessa [JoB12].

Yhteenvetona ulkoisen ympäristön muutokset ovat tutkimuksessa käytetty nimitys ilmiöille, joissa tarkasteltavan kohteen ulkopuolisessa ympäristössä tapahtuva muutos voi aiheuttaa kohteen laadussa havaittavia muutoksia ilman, että kohteeseen tehdään muutoksia. Kohteella tarkoitetaan ohjelmistotuotannon prosessia tai jotakin ohjelmistotuotannon prosessien tuotosta, kuten tietojärjestelmää tai dokumenttia. Jotkin tällaiset tilanteet saattavat ensisilmäyksellä muistuttaa teknistä velkaa. Niissä on kuitenkin luonne-ero: Kun tekninen velka on organisaation oman – enemmän tai vähemmän tietoisien – päätöksenteon tulosta, ulkoisen ympäristön muutos on organisaatiosta riippumattomampi, laajemman ekosysteemin muutos, jonka ennakkointi vaatii enemmän ennustamiskykyä.

4.1.5 Tietojärjestelmän kriittisyys

Komponentti tai toiminto, kuten tietojärjestelmä, voidaan määritellä kriittiseksi, mikäli sen vikaantuessa seuraukset ovat erityisen vakavat verrattuna sellaisen toiminnon vikaantumiseen, jota ei tulkita kriittiseksi [Cam20, Inv20, HiC10]. Kirjallisuudessa tunnetaan ainakin neljä kriittisyystyyppiä: turvallisuuskriittisyys, toimintakriittisyys, liiketoimintakriittisyys ja tietoturvakriittisyys (taulukko 5, s. 44).

Kriittisyystyyppi	Merkityksen kuvaus
Turvallisuuskriittinen (Safety-Critical)	Häiriö voi johtaa hengen menetykseen, vakavaan henkilövaurioon tai vahingoittaa ympäristöä [HiC10]
Toimintakriittinen (Mission-Critical)	Häiriöstä voi seurata merkittävä kriittisen infrastruktuurin tai datan menetys [HiC10]. Äärimmäisen tärkeä organisaatiolle, jollekin toiminnalle tms. [Cam20] Häiriöstä seuraa jonkin toiminnan pysähtyminen, esimerkkinä sähköverkko. [Inv20]
Liiketoimintakriittinen (Business-Critical)	Häiriöstä voi seurata merkittäviä aineellisia tai aineettomia vahinkoja liiketoiminnalle [HiC10]. Liiketoiminnan onnistumisen kannalta välttämätön asia. [Cam20]
Tietoturvakriittinen (Security-Critical)	Häiriöstä tai varkaudesta voi seurata sensitiivisen datan menetys [HiC10]

Taulukko 5. Tietojärjestelmien kriittisyystyypit

Haastatteluissa kriittisyys nähtiin ominaisuutena, joka voi vaikuttaa päätöksenteon kautta siihen, missä määrin resursseja allokoidaan. Sillä puolestaan voi olla yhteys laatuun esimerkiksi käytettäessä vähemmän aikaa ja kokemattomampia ihmisiä asioiden tekemiseen.

“Todella low cost – näyttäytyt yleensä siinä, että on esimerkiksi vähemmän osaavia tekijöitä. Prioriteetti näkyy sitten siinä, että jos ollaan tekemässä jotain liiketoimintajärjestelmää, ja ensisijainen draiveri on matala kustannus, niin ei se voi olla prioriteetissa kovin korkealla. Jos sillä olisi liiketoiminnallisesti kriittinen rooli mitä tehdään, silloin sillä olisi myös prioriteettiä, ja matala kustannus ei olisi se ykköskriteeri.” (henkilö C)

Kriittisyyden taso määritellään usein totuusarvoisena eli kahden arvon luokitteluasteikkona: järjestelmä joko on tai ei ole kriittinen [HiC10, Cam20, Inv10]. Tällainen asteikko voi vastaavasti ohjata järjestelmää koskevaa päätöksentekoa kahteen suuntaan.

Hinchey ja Coyle esittävät kriittisyyden ja laadun välille riippuvuuden. He määrittelevät termin Evolving Critical System (ECS) tarkoittavan tietojärjestelmää, jonka laatu ei saa kärsiä järjestelmän elinkaaren missään vaiheessa, kuten jatkokehittäessä tai operoitaessa sitä [HiC10]. Esimerkiksi jos järjestelmän kriittisyyden luonteeseen liittyy korkean saatavuuden vaatimus, se voi olla riittävä perustelu investoida resursseja siihen, että järjestelmälle toteutetaan katkottomien versiopäivitysten mekanismi.

Toinen esimerkki tilanteesta, jossa kriittisyyttä käytetään hyvin eksplisiittisesti tavoitellun laatutason määrittelemiseen, löytyy Suomen Julkisen hallinnon tietohallinnon neuvottelukunnan (JUHTA) suosituksesta JHS 212 ”ICT-palvelujen palvelutasonhallinta (SLM)”. Se määrittelee kriittisyydelle esimerkinomaisesti järjestysasteikon, jossa on neljä luokkaa: alhainen, normaali, tärkeä ja kriittinen. Järjestysasteikkoa hyödynnetään JHS 212:ssa määri-

teltäessä sopimustasolla palveluille eriasteisia saatavuustavoitteita ja päätettäessä, missä järjestyksessä häiriöitä tulee ratkaista tilanteessa, jossa useampi palvelu vikaantuu samanaikaisesti [JHS212].

Laatustandardi ISO/IEC 9000:2015 tunnistaa kaksi eri tyyppiä attribuuteille, joita tarkasteltaviin objekteihin voidaan liittää. Attribuutti voi olla joko objektiin luontaisesti (*inherent*) kuuluva, kuten silmien väri, tai määrätty (*assigned*) ominaisuus, kuten tuotteen hinta [ISO90]. Tätä määrittelyä käyttäen todetaan, että kriittisyys on luonteeltaan määrätty ominaisuus, eikä järjestelmän luontainen ominaisuus. Saman tietojärjestelmän tai ohjelmistotuotteen kriittisyys voi vaihdella eri ympäristöissä tai ajan kuluessa käyttötarkoitusten muuttuessa. Esimerkiksi organisaatio voi ottaa käyttöön uuden tietojärjestelmän, jonka vastuulle siirretään jokin kriittinen toiminto vanhalta tietojärjestelmältä, joka on lähestymässä elinkaarensa loppua. Tämä voi vaikuttaa siten, että vanhan järjestelmän kriittisyys laskee.

4.1.6 Tulevaisuudenkestävyys

Klassinen IT-alan esimerkki tulevaisuudenkestävyyden kategoriaan luettavasta ongelmasta oli Vuosi 2000 -ongelma. Suuri määrä maailmalla käytössä olevista ohjelmistoista oli toteutettu siten, että tilan säästämiseksi vuosiluvut tallennettiin ja käsiteltiin vain kahdella viimeisellä luvulla suuremman lukumäärän sijasta. Vuosi 2000 -ongelmasta aiheutui kovasti töitä ja investointeja vuoden 2000 lähestyessä [Lav19].

Haastateltavat E ja F kuvasivat tapauksia, joissa tietojärjestelmän elinkaaren myöhemmissä vaiheissa on tullut vastaan yllätyksiä, jotka olisi ollut arvokasta pystyä ennustamaan jo järjestelmän hankintavaiheessa. Tapausten luonne on ollut kuitenkin sellainen, että sellaisten mahdollisuus ei ole juolahtanut mieleen hankintavaiheessa:

“On järjestelmiä, joissa tietokantarakenne on täysin toimittajan varassa. [Meillä asiakkaalla] ei ole mitään kuvausta. Jos halutaan jotain rajapintaa ja tietoa ulos, toimittaja rakentaa sen rajapinnan, ja se maksaa tosi paljon: toimittaja osaa kyllä hinnoitella sen. Esimerkiksi henkilöstöhallinnon palkkajärjestelmän toimittaja ilmoitti, että vuoden lopussa tuki päättyy, ja täytyy sen vuoksi päivittää täysin toiseen järjestelmään. – Hankintavaiheessa pitäisi olla selkeä vaade, että järjestelmässä on riittävä API sekä tietomallin kuvaus jotta sieltä saa datan ulos. Muutoin tietojärjestelmästä datan ulos saaminen saattaa helposti maksaa parin vuoden käytön verran.” (henkilö F)

“Tyypillinen, mikä tapahtuu aina joka ikisessä projektissa, on se, että eihän kukaan tarjoa koodia niin pitkälle mietittynä, että... tavallaan niinku, tulee yllätyksiä, että ‘voi vitsi, eikö tätä pystykään vaihtamaan helposti?’ Sanotaan vaikka, että haku ei toimi – on vain [todella] hidas ja indeksoi huonosti. Sitten mietitään että ‘no, vaihdetaan haku’, mutta vastaus onkin, että ei olekaan ihan helppoa vaihtaa.” (henkilö E)

“Sitäkin tapahtuu tosi paljon, että otetaan joku kirjasto käyttöön, käytetään siihen kaksyhtä päivää, että

saadaan se toimimaan. Sitten yllähuomenna [meille] ilmoitetaan, että nyt on tuki päättynyt, jonka jälkeen tietoturvaihmiset sanoo, että tämä pitäisi saada nyt pois.” (henkilö E)

Haastattelussaan henkilö B kertoi tarinan finanssialan useasta eri toimijasta, jotka olivat keränneet dataa hieman eri tarkkuustasoilla. Se aiheutti myöhemmin ongelmia yritettäessä yhdistää dataa ja rakentaa sen varaan eräänlaista keskitettyä API-palvelua. Vuosien ajalta kerättyyn dataan ei myöskään ollut mahdollista jälkikäteen lisätä jo alunperin puuttunutta tarkkuutta. Haastatteluissa nousi esille myös muita vastaavan tyyppisiä dataan liittyviä haasteita:

“Pitkäikäisissä IT-järjestelmissä datan mallinnuksen laatu on mun mielestä paljon suuremmassa roolissa kuin minkään muun asian laatu. – – Aika helposti voit korjata järjestelmätason ongelmia jos ne liittyy siihen, että joku rajapintatoteutus tai integraatio on huonosti koodattu – viime kädessä siten, että koodaat sen alusta uusiksi. Mutta jos sulla on kymmenen vuoden datat sellaisessa huonosti mallinnetussa tietovarastossa, niin sitäpä et noin vaan korjaakaan.” (henkilö C)

”Datan elinkaaresta: jotain dataa tarvitaan hyvin kaukaakin historiasta, mikä vaikuttaa tarpeeseen säilyttää dataa, sekä siihen käyttöliittymää.” (henkilö F)

“On ihan käytännönkin kokemusta siitä, että siinä vaiheessa kun tehdään jotain migraatioprojektia, havahdutaan siihen että – hetkinen – laki sanoo, että pitää säilyttää [dataa] 10 vuotta – – Datan elinkaarta ei ole mietitty omana elinkaarenaan, vaan se tulee yllätyksenä, kun tehdään joku migraatioprojekti. Kun joku järjestelmä vaihdetaan, niin tajutaan, että – hetkinen – nää datat pitää saada pysymään tallessa. Pitää olla luettavissa, pitää olla käytettävissä. Eikä olla mietitty sitä, että datan esitystapa muuttuu migraatiossa. Musta tuntuu että yritykset ei hirveästi ole tiedostaneet tätä vielä.” (henkilö D)

Tulevaisuudenkestävyyttä analysoitaessa on hyvä ymmärtää eteen- ja taaksepäin yhteensopivuus, mutta kyse ei ole täysin samasta asiasta:

“Joku asia, joka on ollut mallinnuksessa vaikea ja vaikea muuttaa, saattaa olla laajennettavissa – – taaksepäin ja eteenpäin yhteensopivasti – – mutta ei fundamentaalilla tavalla muutettavissa. – – Voi mennä myös niin päin, että ollaan valittu kompleksi esitystapa asialle ja se kompleksius tekee siitä muuttamisesta vaikeaa, kun mallinnuksellisesti yksinkertaisempi tapa olisi voinut olla työläämpi toteuttaa softaan, mutta se olisi jättänyt enemmän avoimia polkuja tulevaisuudelle. Kalliiksi tulevat päätökset ovat usein sellaisia, jotka sulkevat ovia, eivätkä avaa niitä.” (henkilö C)

Ryan ja Ur Rehman määrittelevät tietojärjestelmän tulevaisuudenkestävyyden olevan kyky ennustaa ja jossain määrin varautua mahdollisiin tulevaisuuden muutoksiin. He tekevät eron tulevaisuudenkestävyyden ja sellaisen järjestelmän mukautuvuuden välillä, jossa järjestelmän mukautuvuus määritellään ei-toiminnallisilla vaatimuksilla, kuten skaalautuvuudella [RyU18].

Teknisen tulevaisuudenkestävyyden lisäksi yksi keino on kerätä eräänlaista jatkuvuusrahastoa tulevaisuuden ennalta-arvaamattomampien muutosten varalta:

“Joka budjettiinhan suunnitellaan contingency-osuus – – Sen osuutta mä olen kyllä kasvattanut joka vuosi. Välillä sitä tuntuu olevan vaikea perustella.” (henkilö E)

”Mieluummin mä käyn sanomassa [budjettia hakiessani] että ‘tää maksaa puoli millia’ kuin että yritän optimoida ja sanoa että ‘tää maksaa 385 tonnia’ – ja sit mä olen ihan ongelmissa sen kanssa, koska enemmän tulee huutoa siitä, että sä yrität sitten selittää sitä 385:stä sinne puoleen milliin, kuin että sä

kerran sanot, että puoli millia, ja sitten pysytään sen sisällä. – – Klassinen moka: saat tarjouksen, verkkopalvelun rakentaminen maksaa sata tonnia. Ja lähdet hakemaan sillä budjettia että ‘tää on nyt sata tonnia’. Nykyäänhän alkaa olla silleen että sen rakentaminen on yksi, sitten on palvelumuotoilu, voi olla jotain asiakaskyselyitä, tietoturvatestauksia, kuormitustestauksia, ja sitten on infrastruktuuria, jatkokehitysjuuttuja, joku lisäpaketti siihen kun se on julkaistu, ja ja – mitä kaikkea meillä voi olla – ja sitten meillä on vielä se contingency. Niin tätä on joskus haukkoiltu että ‘sää saat tarjouksen sadasta tonnista ja sä pyydät kaksisataaviiskyt tonnia rahaa?’ Nimenomaan. Ei se ole enää yksi ainoa asia se että joku koodaa jonkun asian. Se on yksi osa koko kokonaisuutta.” (henkilö E)

Yhteenvetona haastatteluissa nousi esiin useita erilaisia tapauksia, joissa tietojärjestelmää hankittaessa ei ole osattu huomioida tulevaisuuden tarpeita riittävästi. Tällaiset tilanteet ovat tuottaneet myöhemmin laatuongelmia, kuten teknisiä haasteita, tai luoneet toimittajaloukun, jonka kautta toimittaja on pystynyt vaatimaan tarvittavista muutoksista suhteettoman kovaa hintaa. Myös tämä voidaan nähdä laatuongelmana, koska järjestelmän odotetut kustannukset eivät yllättäen vastaakaan suunniteltua.

4.1.7 Omistajuus ja proaktiivisuus

Omistajuus on tässä tutkimuksessa valittu termi kuvaamaan vastuuta, johon voi liittyä eksplikoitu tulostavastuu tai suoritusvastuu, mutta joka voi olla myös jokin sen luonteinen tapaus, missä vastuun voi ajatella olevan olemassa ilman erillistä määrittelyäkin. Proaktiivisuus on oma-aloitteisuutta toimia jonkin asian suhteen.

Haastatteluissa nousi esille tapauksia, joissa laatuongelmia on päässyt syntymään omistajuuden tai proaktiivisuuden puutteen vuoksi. Henkilöiden A ja B esimerkeissä ketään ei ole riittävän eksplisiittisesti vastuutettu jonkin tietyn asian hoitamisesta. Henkilön D esimerkissä tietyn asian omistajuus on hyvinkin selkeästi määritelty tietylle organisaation toiminnolle, joka myöskin tietää oman vastuunsa ja yrittää hoitaa sen, mutta ei onnistu siinä halutulla tavalla:

“Iso syy on omistajuuden puute: ei ole ketään organisaatiossa, joka vastaisi integraatioarkkitehtuurista.” (henkilö B)

“On varmaan ollut enemmänkin yksittäisistä verkkopuolen henkilöistä kiinni, kuinka he suhtautuu tilanteisiin. Mahdollinen tietoturveysikkö, jos sellainen on olemassa, kuinka he suhtautuu tilanteisiin. – – Se, mitä useimmiten puuttuu, on oman organisaation työtehtävien taskittaminen. Hyvin harvoin on niin, että se mikä riittää on, että soitat toimittajalle ja [delegoit:] ‘hoidatko alusta loppuun’. Tarvitaan kaikenlaista ympärille. Kun perusasioita ei ole valmiiksi taskitettu omassa päässä, aina törmätään [siihen, että]: toi on unohdettu, toi on hoitamatta.” (henkilö A)

”Järjestelmäarkkitehtien painoarvo organisaatiossa voi olla suhteessa liian iso. Tai on funktio olemassa, mutta ei ole voimaa muuttaa toimintaa. – – Olen nähnyt vähän ton tyyppistä siinä, että joku on älähtänyt, että ‘teidän olisi pitänyt pyöryttää tuo tämän meidän arkkitehtuuriprosessin kautta, jotta oltaisi nähty voitaisiinko tämä tehdä jotenkin hienommin ja pitempää aikajaksoa ajatellen.’ – – Konkreettinen esimerkki: tein projektia eräälle asiakkaalle, joka oli osa isompaa konsernia. Niillä oli tosi itsenäinen IT eri osastojen välillä. Ne oli päättäneet jo teknologian, millä tehdään, ja asiaa oltiin teke-mässä heidän hyväksymällään tavalla. Oltiin [toteutuksessa] jo hyvin pitkällä etenemässä. Kunnes

sitten [konsernin] ylätasolta arkkitehdit ärähtivät, että 'hei, tämä ei täytä vaatimuksia, meidän vaatimuksena on, että käytetään tuotetta X eikä muita tuotteita'. Tavallaan [se arkkitehtuuritiimi] oli – – horisontaali kerros, joka läpäisi kaikki vertikaalit osastot [ja] joka teki kaikille tällaisia päätöksiä. Kukaan asiakkaalla ei ollut tietoinen siitä, että sellainen oli olemassa. Se oli niin hämmäntävä tilanne, etten oikein ymmärtänyt miten sellainen pääsee syntymään.” (henkilö D)

Edellisissä esimerkeissä omistajuus ja sen puute ilmenevät yhden ja saman organisaation sisällä. Tosin henkilö D:n esimerkin organisaatiotyyppinä ollut konserni alkaa olla kooltaan ja monimutkaisuudeltaan sellainen, että se voi lisätä tarvetta määritellä eri toimintojen mandaatit selkeämmin kuin yksinkertaisemmassa organisaatiossa.

Henkilö A nostaa esille sen tyyppisen omistajuuden puutteen, joka ylittää organisaatorajat. Kyse on tilanteista, joissa palvelua tilaavalla organisaatiolla on päässyt syntymään vetovastuurooli, vaikka näin ei ehkä haluttaisi olevan. Kun tällaisessa tilanteessa palvelun toimittajalta odotetaan oma-aloitteista toimintaa, sitä ei tapahdu. Työ jää tekemättä, ellei asiakas sitä erikseen vahdi ja tilaa:

“Ensimmäinen draivi lähtee asiakkaiden vaatimuksista. Jos asiakkaat ei vaadi, ei ohjelmistotalokaan lähde tekemään uutta versiota vain siksi, että se olisi heidän mielestään järkevä vaihtoehto – – Asiakkaat useimmiten on se, mikä sanelee sen, että 'hei nyt meillä on tarvetta päivittää'. Ja kovin monesti tämä tapahtuu vasta siinä vaiheessa kun käytetty versio tulee end-of-life -tilaan, sitten vasta aletaan harkitsemaan, että jotain pitäisi tehdä. Pitäisi olla [palveluntoimittajalle] täysin selvää, että kun tulee joku uusi versio [alustasta], jossain vaiheessa asiakkuudet tulevat siirtymään siihen” (henkilö A)

"Jos saisi edes sen pienen asian sovellustoimittajille juntattua, että 'seuraa CVE-tietokantaa käyttämiäsi komponenttien osalta', oltaisiin tietoturvan näkökulmasta paljon paremmassa tilanteessa.” (henkilö A)

Henkilön A kuvaamissa tapauksissa on kyse selkeästi määriteltävissä olevista asiakokonaisuuksista, kuten säännöllisistä tietoturvapäivityksistä. Tällaiset olisi mahdollista sopia ja kuvata esimerkiksi vastuunjakomatriisina tai ottamalla ne osaksi suoritettavia prosesseja:

“[On tärkeää että olisi] prosessit mietittynä erilaisiin tilanteisiin, vakioasioihin ja poikkeuksellisiin. Eli jos sulla tulee tietty heräte jostain, [prosessi kertoisi] mitä pitää tapahtua seuraavaksi, että prosessi etenee. Prosessin työstäminen on ehkä yksi tärkeimpiä asioita. Se voi olla äärimmäisen yksinkertainen, ei tarvii olla mitään ITIL-mallia tai vastaavaa. Riittää, että on selkeästi vain seipitetty, että jos joku havaitsee jonkun – vaikka bugin – järjestelmässä, niin siitä kirjataan bugiraportti. Kuka ottaa katsoakseen onko se bugi vai ei, miten siitä eteenpäin. – – Selkeästi sovittu miten se toimii, ketkä tulee mukaan, kuka ottaa vastuuta. Turha jahkaaminen jää pois. Prosessejakin pitää toki evaluoida. Prosessimaisuus, sanotaan yli 5 hengen firmoissa, alkaa olla tärkeitä. – – Näkisin niin, että mikä tahansa yritys [onkaan] kyseessä, niin [ota käyttöön] yksinkertainen tikettijärjestelmä tai flow tai vastaava. Tee taskeja tiettyihin tarpeisiin, ihan perus-arkisiin asioihin. Silloin kun sulla on taskitettuna ne tehtävät, tiedät prosessimallisesti ne stepit mitä sun on tehtävä, että asia menee maaliin. Silloin voit hyödyntää sitä olemassaolevaa mallia ja tiedät että se toimii. Et unohda asioita. Ettei sulla ole taas joku projektipäällikkö vetämässä kaikkea kasaan, että missä me mennään, kasaamassa yhteen missä meillä on keskeneräisiä asioita. Koska ei ne välttämättä tule ilmi ellei niitä ole kerrottu jo heti alussa että mitä kaikkea sun pitää hoitaa. ” (henkilö A)

Proaktiivisuuden tarve voi olla myös vaikeammin määriteltävä asia. Henkilö E nostaa esille palveluita hankkivan tahon roolissaan sellaisen proaktiivisuuden odotuksen, jossa hän kaipaa palvelun toimittajalta eräänlaista innovointitukea tai oma-aloitteista otetta auttaa varmistamaan, että tehdään järkeviä töitä:

“Toimittajat kun on pitkään samassa [roolissa, työssä] niin ne saattaa vähän silleen laiskistua, että niiltä ei tule enää yhtään ideaa koko hommaan. Sitten kun on lyhyempiä sopimuksia, niin molemmilla on semmonen draivi kehittää sitä yhteistyötä. Myöskin se asiakas vähän laiskistuu, ‘tossahan on hyviä jätkeä, ne on hoitanut tätä jo viisi vuotta’, mutta ei tuu katottua sitten, että teidänköhän me enää ihan oikeita asioita. Haen itse monitoimittajaympäristöä, koska siinä tulee positiivinen haastaminen. Toisella toimittajalla voisi olla vähän nälkää saada kaverinkin siivua, voisi ihan hyvin tehdä vaikka tota ja.. Toinenkin joutuu vähän tsemppaamaan.” (henkilö E)

Tällaista asiakkaan odotusta palveluntoimittajan proaktiivisuudelle voi olla hankalampi koettaa määritellä kovin eksplisiittisesti, kuten vastuunjakomatriisiksi sopimuksiin. Proaktiivisuuden merkitys asiakkaan kokemaan palvelun laatuun voi olla kuitenkin niin merkittävä, että ilmiön ymmärtäminen voi olla molemmille osapuolille oleellista.

Näiden esimerkkien perusteella omistajuuden ja proaktiivisuuden puutteilla voi olla isoja vaikutuksia ohjelmistotuotannon laatuun. Ne voivat aiheuttaa muun muassa oleellisesti laatuun vaikuttavien tehtäväkokonaisuuksien kokonaan tekemättä jättämistä tai niiden tekemistä vain osittain. Syitä omistajuuden haasteisiin voivat olla ainakin muutosvastarinta, sisäisen viestinnän epäonnistuminen, organisaatiokulttuuritasolla tarvittavan muutoksen epäonnistuminen, prosessien tasolla tarvittavien muutosten epäonnistuminen, organisaation hallinto- tai johtamistasolla tapahtunut liian epäselvä vastuutetun asiakokonaisuuden määrittely tai kokonaan tekemättä jäänyt vastuunjako.

4.1.8 Kognitiiviset vinoumat

Kognitiiviset vinoumat olivat tuttuja osalle haastateltavista. Kaksi haastateltavista mainitsi ankkuroidivaikutuksen ilmiönä, johon olivat törmänneet työelämässä. Siinä ihminen painottaa päätöksenteossa liikaa ensimmäistä hänelle tarjottua tietoa:

“Älä koskaan aluksi kerro mitään numeroa koska se palaa niiden verkkokalvoille. Jos heti ensimmäisellä tapaamisella sanot jonkun luvun, se on aina se mikä muistetaan, ei lopulta [sitä] että [mihin tietoihin se perustui]” (henkilö E)

Henkilö F mainitsee törmänneensä sitoutumisen eskalaatioon:

“Kyllä, tietojärjestelmiä, joihin on laitettu paljon rahaa ja joista halutaan siksi pitää kiinni, vaikka olisi ollut järkevämpi uusia ne jo aikaisemmin.” (henkilö F)

Henkilö E tuo esille kognitiivisen vinouman, jossa omat tunteet, kuten toisesta henkilöstä pitäminen, vaikuttavat päätöksentekoon:

“Se mikä myös vaikuttaa on henkilökemia. Jos ihmisen kanssa tulee juttuun ja tuntuu, että [hän] ymmärtää [mitä tarvitaan], jos hänen tarjous [sen jälkeen] on huono, sitä on vaikea olla hyväksymättä.” (henkilö E)

Organisaatioiden päätöksentekoprosesseissa voidaan pyrkiä huomioimaan kognitiivisten vinoumien mahdollisuus. Esimerkiksi saksalaisessa RWE:ssä analysoitiin päätöksentekoa, tunnistettiin useita eri päätöksentekoon vaikuttavia vinoumia ja otettiin käyttöön niitä huomioivia ja niiden eliminointiin pyrkiviä mekanismeja. Keskeinen muutos oli vaikuttaa organisaatiokulttuurin kautta nostamalla kognitiiviset vinoumat esille luonnollisena ja ymmärrettävänä ilmiönä, mutta samalla myös ilmiönä, jota halutaan jatkossa alkaa kontrolloida. Muutosvaikutuksiltaan isoimpiin päätösprosesseihin tuotiin mukaan uusi rooli, “paholaisen asianajaja”, jonka tehtävänä oli tuoda mukaan kriittistä tarkastelua ja pyrkiä tunnistamaan ja nostamaan esille päätöksentekoprosessiin mahdollisesti vaikuttamaan päässeet kognitiiviset vinoumat [GHW17].

4.2 Ilmiöiden esiintyminen ohjelmistotuotannon viitekehyksissä

Tässä luvussa kuvataan, missä määrin valitut viitekehykset tunnistavat löydetty ilmiöt. Kaikki paitsi ISO/IEC 25000 olivat PDF-muotoisia. ISO/IEC 25000:n kohdalla dokumentit koostuivat huonolaatuisista paperisten dokumenttien skannauksista, joten etsintä suoritettiin lukemalla dokumentit manuaalisesti. Muiden dokumenttien kohdalla hakumenetelmänä käytettiin tekstihakuja.

Koska viitekehykset ovat englanninkielisiä, ilmiöiden nimet ja niihin läheisesti liittyviä termejä käännettiin ensin englanniksi. Näin saaduista sanoista otettiin lyhyemmät osamerkkijonot, joilla tekstihaut suoritettiin kirjainkorkoriippumattomasti. Taulukko 6 (s. 51) esittää tekstihakutermit, joita tutkimuksessa käytettiin. Tekstihakujen tuloksena löytyi valtava määrä vääriä positiivisia tuloksia, mikä oli odotettavissa. Esimerkiksi hakutermi ”critic” löysi PMBOK-viitekehyksestä kaikki kohdat, joissa puhuttiin projektinhallinnan kriittinen polku -käsitteestä. Tämän vuoksi hakutulokset täytyi käydä manuaalisesti läpi.

Ilmiö	Käytetyt tekstihakutermit
Niukkuus	cost, cba, benefit, opportunit, risk, scarc, pauc
Tiedon määrä ja laatu	inform, data, decision
Tekninen velka	debt
Ulkoisen ympäristön muutos	life, time, change, extern, impact
Tietojärjestelmän kriittisyys	critic
Tulevaisuudenkestävyys	future, contingen
Omistajuus ja proaktiivisuus	own, account, respons, raci
Kognitiiviset vinoumat	cognit, bias, psych

Taulukko 6. Ilmiöiden viitekehyksistä etsimisessä käytetyt tekstihakutermit

Taulukko 7 esittää hakutulosten läpikäynnin tuloksena muodostetun käsityksen siitä, kuinka kattavasti ilmiöt ovat viitekehyksissä edustettuina.

Ilmiöt	ISO/IEC 250nn	COBIT	IT4IT	TOGAF	SWEBOK	PMBOK
Niukkuus		X	X	X	X	X
Tiedon määrä ja laatu	1	X	X	X	X	X
Tekninen velka						6
Ulkoisen ympäristön muutos		X	3	4	2	X
Tietojärjestelmän kriittisyys		X	X	X	X	X
Tulevaisuudenkestävyys		5		X		
Omistajuus ja proaktiivisuus		X	X	X	X	X
Kognitiiviset vinoumat		7				

Taulukko 7. Ilmiöiden esiintyvyys valituissa ohjelmistotuotannon viitekehyksissä

Solussa oleva arvo jäsentää viitekehykset eri ryhmiin seuraavasti: Merkki X tarkoittaa, että ilmiö esiintyy viitekehyksessä selkeästi ja isommassa merkityksessä kuin esimerkiksi pelkästään lyhyenä kommenttina sanastossa. Numerot 1–7 kertovat, että ilmiö esiintyy vähintään yksittäisessä kohdassa viitekehystä, esimerkiksi sanastossa, mutta selvästi pienemmässä

mittakaavassa kuin X-tasolla. Numeroiden merkityksestä on annettu tarkemmat kuvaukset alla:

- 1: Ei käsittele päätöksentekoa prosessina tai juuri siihen liittyvää tietoa, mutta tarjoaa datan laatumallin, jota voidaan hyödyntää myös päätöksentekoa tukevan tiedon laadunvarmistuksessa
- 2: Impact analysis ja Anticipating change -näkökulmista
- 3: Reaktiivisesti, kun muutos on jo päässyt tapahtumaan ja havaitaan esimerkiksi häiriönä. Palveluiden operoinnin tasolla Tapahtumanhallinta-prosessissa
- 4: Muutokset huomioidaan Architecture Change Managementin kautta, jos ovat vaikutukseltaan riittävän laajoja
- 5: Contingency planning
- 6: Mainitsee termin Technical debt, mutta vain lyhyesti sanastossa ja Agile Pain Points -taulukossa
- 7: Mainitsee riskianalyysin yhteydessä: ”verify that estimations were properly calibrated and scrutinized for bias”

Tyhjä solu tarkoittaa, ettei ilmiö esiinny viitekehyksessä lainkaan. Jos hakutermeillä on löytynyt tuloksia, ne ovat käsitelleet jotakin muuksi tulkittua ilmiötä.

5 Pohdinta

Tutkimuksen tarkoituksena oli kartoittaa laatua haastavia ilmiöitä ohjelmistotuotannon päätöksenteossa. Konkreettisenä tavoitteena oli käsitteellistää kartoituksen kautta esiin nousevia ilmiöitä ja tuoda niiden avulla uutta näkökulmaa laadusta käytävään keskusteluun. Lisäksi tavoitteena oli selvittää missä määrin ilmiöt esiintyvät tietoperustaan valikoiduissa ohjelmistotuotannon viitekehyksissä.

5.1 Laatua haastavien ilmiöiden tarkastelua

Haastatteluilla tunnistettiin joukko ilmiöitä, jotka ryhmiteltiin ja abstrahoitettiin sisällönanalyysin yhteydessä kahdeksan kokonaisuuteen: niukkuus, tiedon määrä ja laatu, tekninen velka, ulkoisen ympäristön muutos, tietojärjestelmän kriittisyys, tulevaisuudenkestävyys, omistajuus ja proaktiivisuus sekä kognitiiviset vinoumat. Tämä lista vastaa tutkimuksen tutkimuskysymykseen TK1.

Voidaan huomata, että ilmiöt muodostavat hyvin heterogeenisen joukon. Voiko esimerkiksi ulkoisen ympäristön muutoksella ja kognitiivisella vinoumalla olla mitään muuta yhteistä kuin se, että ne molemmat voivat vaikuttaa ohjelmistotuotannon laatuun? Ilmiöitä lähdettiin tarkastelemaan tämä kysymys mielessä, ja sen tuloksena syntyi joukko akseleita tai mittareita (taulukko 8, s. 54).

Ilmiöihin liittyy muun muassa keskeisesti tietoisien ja tiedostamattoman akseli. Teknistä velkaa esimerkkinä käyttäen organisaatio voi olla hyvin tietoinen siitä velan määrästä mitä on kerrytetty. Velkaa saatetaan arvottaa rahana ja päättää tulevien tilikausien budjetoinnissa sen takaisinmaksun tahdista ja siihen sijoitettavasta rahasta. Toisaalta organisaatio voi olla täysin tietämätön koko ilmiön olemassaolosta.

Taulukkoa voi käyttää pohtimalla jotakin kiinnostuksen kohteena olevaa ilmiötä, käymällä listaa läpi tarkistuslistamaisesti ja esittämällä kysymyksiä kuten: ”Liittyykö tähän jotain selkeästi ennakoitavia asioita? Entä voiko tähän liittyä jotain yllätyksellistä?” tai ”Mitkä tämän vaikutukset ovat lyhyellä aikavälillä? Entä pitkällä aikavälillä?”.

tietoinen	tiedostamaton
faktatieto	arvio/arvaus
ennakoitavuus	yllätyksellisyys
riskien eliminointi	riskien ottaminen
tiedon jakaminen ja läpinäkyvyys	tiedon jakamattomuus
lyhyt aikaväli	pitkä aikaväli
vakiintuneiden käytäntöjen noudattaminen	vakiintuneiden käytäntöjen kyseenalaistaminen

Taulukko 8. Akseleita laatua haastavien ilmiöiden tutkailuun

Vastaus tutkimuskysymykseen TK2 eli ilmiöiden esiintyminen viitekehyksissä (taulukko 7, s. 51) tuo esille ehkä hieman yllättävän seikan: niinkin tutkitut käsitteet kuin tekninen velka ja kognitiiviset vinoumat eivät ole tässä viitekehysten joukossa edustettuina missään muussa kuin PMBOK:ssa, ja siinäkin vain hyvin lyhyesti sanastossa ja Agile Pain Points and Troubleshooting Possibilities -kohdassa.

Tämän löydöksen perusteella voidaan todeta ainakin seuraavaa: Mikäli organisaatio on laa- timassa omaa IT:n hallinnoinnin ja johtamisen järjestelmäänsä ja valitsee sen pohjaksi pel- kästään jonkin osajoukon tässä tutkimuksessa tarkastelluista viitekehyksistä, kaksi hyvin merkittävää laatuun vaikuttavaa käsitettä, tekninen velka ja kognitiiviset vinoumat, voivat jäädä organisaatiolta tiedostamatta, ellei niitä huomata erikseen räätälöidä mukaan. Yleiste- tymmin voidaan ajatella, että jos luottaa vahvasti viitekehyksiin, malleihin tai vaikkapa tar- kistuslistoihin, saattaa lakata tiedostamasta, että niiden ulkopuolelle voi jäädä jotain oleel- lista. Tämäkin on eräänlainen kognitiivinen vinouma, selviytymisharha.

Menetelmä, jolla selvitettiin ilmiöiden esiintymistä viitekehyksissä, palveli tätä tutkimusta riittävällä tarkkuustasolla. On kuitenkin hyvä tiedostaa, että se on virhealtis ja subjektiivinen. Sen vuoksi sitä ei voi esimerkiksi käyttää osoittamaan aukottomasti, ottaako jokin vii- tekehyksistä kantaa tiettyyn ilmiöön vai ei. Virhealttius on seurausta työvaiheesta, jossa hakutuloksia analysoitiin yksi kerrallaan manuaalisesti. Manuaalisesti tehty työ on aina al- tis inhimillisille virheille. Subjektiivisuus puolestaan seuraa siitä, miten käsitteiden merki- tykset tulkitaan ja miten käsitteet samastetaan.

Esimerkiksi tarkasteltaessa tulevaisuudenkestävyyttä, voidaanko COBITin virkkeen "Use the monitoring results to implement improvements and ensure that future budgets are more

accurate" ajatella liittyväksi tulevaisuudenkestävyyden käsitteeseen? Entä liittyykö se riittävän vahvasti siihen samaan tulevaisuudenkestävyyden käsitteeseen, kuin TOGAFin arkitekhdin roolin vastuuta kuvaava ”To ensure that the architecture is technically coherent and futureproof.” ? Tämä ja vastaavat kysymykset rajautuivat tämän tutkimuksen ulkopuolelle. Niiden osalta esitetään jatkokehitysideana viitekehysten yhdistelmäontologian määrittely, joka kuvataan tarkemmin jatkokehitysideat-luvussa.

Niukkuuden yhteydessä puhuttiin organisatorisesta päätöksenteosta ja siitä, että käytännössä päätöksenteko on jotain muuta kuin täydellisen rationaalista. Palataan vielä tarkastelemaan tätä asiaa muutamien eri teorioiden näkökulmasta ja katsotaan, saadaanko selvitettyä niiden perusteella keinoja, joilla päätöksentekoprosessia olisi mahdollista tehostaa.

Organisoituneessa anarkiassa osallistuminen päätöksentekoon on satunnaista, ja päätöksenteon haaste on pitkälti ihmisten aktivoimisen haaste. Inkrementalistisen teorian mukaan päätöksenteossa puolestaan nojaututaan vahvasti aiempiin kokemuksiin siitä, kuinka vastaavissa asioissa on aikaisemmin toimittu. Kahneman ja Tversky esittävät, että päätöksenteossa korostuu vaihtoehtojen yksinkertaistaminen paremmin ymmärrettävään muotoon sekä riskien karttaminen [Hyy12].

Näiden teorioiden kuvaamat ilmiöt voidaan huomioida ja kääntää hyödyksi, jotta päätöksentekoprosessiin saadaan välitettyä tietoa tehokkaammin. Ensiksikin, tiedot vaihtoehtoista kannattaa välittää päätöksentekoprosessiin yksityiskohtaisen tason lisäksi valmiiksi yksinkertaiseksi tiivistettynä. Toiseksi, mukana kannattaa olla valmiiksi tuotettu, huolella valmisteltu riskianalyysi eri vaihtoehtoista. Kolmanneksi, mikäli vaihtoehtoja tarjoavalla taholla on kokemuseräistä tietoa vastaavien vaihtoehtojen valinnoista jossain vertailukelpoisissa aiemmissa tilanteissa, se kannattaa välittää päätöksentekoprosessiin. Neljänneksi, ihmisiä voi koettaa aktivoida tekemään hyvää päätöksentekoa esimerkiksi kuvaamalla miksi päätös on tärkeä.

Kaikki edellä listatut ideat päätöksenteon tehostamisesta ovat viestintää. Viestinnän tärkeys nousi esille myös haastattelujen yhteydessä. Ehkä lopulta kaikkein syvin tutkimuksen taustalla oleva tema on ilmaistavissa muokkaamalla hieman Dick Fairleyn siteerausta:

Quality is free but the real question is if you can persuade decision makers to invest in it.

5.2 Tutkimuksen arviointi

Tutkimuksen arvioinnissa keskeiseksi tarkastelun kohteeksi nousee kysymys tutkimuksen luotettavuudesta. Tässä yhteydessä usein mainittuja käsitteitä ovat reliabiliteetti ja validiteetti, jotka ovat peräisin määrällisen tutkimuksen puolelta. Reliabiliteetilla viitataan tutkimuksen toistettavuuteen eli tulosten pysyvyyteen. Validiteetti puolestaan tarkoittaa, että tutkimuksessa on tutkittu oikeita asioita eli tehty, mitä on luvattu [Kan08]. Laadullisessa tutkimuksessa luotettavuuden arviointi ei kuitenkaan ole yhtä suoraviivaista kuin määrällisessä tutkimuksessa. Esimerkiksi täydelliseen toistettavuuteen on käytännössä mahdotonta päästä, sillä laadulliset tutkimukset ovat aina ainutkertaisia kokonaisuuksia, jotka ovat avoinna erilaisille tulkinnoille.

Tuomi ja Sarajärvi nostavat esille laadullisen tutkimuksen luotettavuuteen liittyvinä tekijöinä muun muassa tutkimuksen johdonmukaisuuden ja riittävän tarkan kuvauksen tutkimuksen toteutuksesta. Lukijalla tulee olla mahdollisuus seurata tutkijan ajatuksenjuoksua ja päättelyä voidakseen arvioida tutkimuksen tuloksia [Tus03]. Vastaavasti Eskola ja Suoranta toteavat laadullisen tutkimuksen luotettavuuden perustuvan tutkijan luotettavuuteen – hänen rehellisyyteensä, tekoihinsa ja valintoihinsa [EsS00]. Siten luotettavuuden arvioinnin tulisi olla jatkuvasti läsnä tutkimusprosessissa ja tehdyt ratkaisut ja valinnat perustella tutkimustekstissä. Huolellisen dokumentaation avulla voidaan varmistaa tutkimuksen toistettavuus sellaisena kuin se laadullisen tutkimuksen piirissä nähdään: mahdollisuutena päästä samoilla menetelmillä samoihin tuloksiin kuin alkuperäinen tutkija [Vil15].

Tämän tutkimusraportin luvussa kolme kuvattiin tutkimuksessa hyödynnetyt menetelmät ja käytännön toteutus. Aineistonkeruvaiheessa tehdyt haastattelut äänitettiin, jotta tutkijan oli analysointivaiheessa mahdollista palata yhä uudelleen haastattelutilanteisiin ja lisätä omaa ymmärrystä tutkimuksen kohteena olevista ilmiöistä. Tulosten yhteydessä tutkija toi esille tekemiään päätelmiä ja tulkintoja, joiden havainnollistamiseksi tekstiin sisällytettiin katkelmia haastatteluaineistosta. Näin lukijalle tarjoutui mahdollisuus arvioida tutkijan ajattelun oikeellisuutta, millä osaltaan pyrittiin vaikuttamaan myönteisesti tutkimuksen luotettavuuteen. Raportoinnissa tavoiteltiin eheää ja helposti seurattavaa kokonaisuutta.

Aineiston riittävyys nousee niin ikään esille esimerkkinä laadullisen tutkimuksen arviointikriteereistä. Laadullisen aineiston optimaalista kokoa on jokseenkin mahdotonta päättää etukäteen [EsS00]. Kananen toteaa osuvasti, että aineisto jatkuu periaatteessa loppumatto-

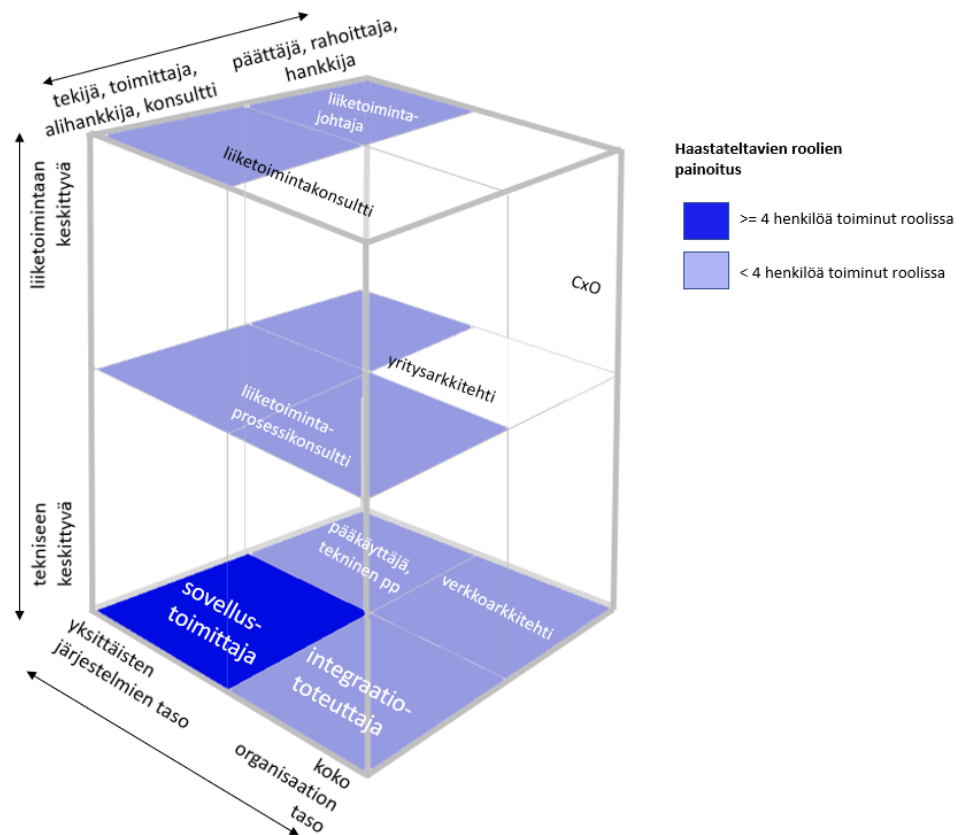
miin, ellei tutkija aseta sille päätepestettä [Kan08]. Riittävyyttä voidaan lähestyä esimerkiksi saturaation eli kylläntymisen käsitteen kautta. Siinä ajatuksena on, että aineistoa on riittävästi, kun se alkaa toistaa itseään [TuS03]. Lopulta riittävyys perustuu kuitenkin aina tehtävään tutkimukseen ja tutkijan harkintaan: aineistoa kerätään sen verran, kuin on tarpeen tutkimuksen kannalta [EsS00]. Aineiston riittävyuden ohella ja osana tutkimuksen arviointia tutkijan tulee arvioida laajemminkin tekemiään aineiston hankintaan ja analysoimiseen liittyviä valintoja [TuS03].

Tässä tutkimuksessa empiirinen aineisto koottiin seitsemällä teemahaastattelulla. Lyhyimmilläänkin reilun tunnin mittaiset haastattelut tuottivat runsaasti tutkimuksen kannalta merkityksellistä aineistoa: kuvauksia kokemuksista ja tilanteista, jotka valottivat laatua haastavia ilmiöitä ohjelmistotuotannon päätöksenteossa. Haastatteluiden rytmittämiseksi valitut teemat ja niitä havainnollistavat kuvat toimivat erinomaisesti tehtävässään keskustelun herräinä. Mitä tulee aineiston riittävyteen, aineiston analysoinnin yhteydessä havaittiin, että neljän haastattelun jälkeen ilmiöt alkoivat kertautua haastateltavien kertomuksissa. Aineistonkeruu päätettiin lopettaa seitsemännen haastattelun jälkeen. Siihen mennessä muodostunut aineisto näyttäytyi tarpeeksi kattavana, jotta siitä oli mahdollista tehdä päätelmiä.

Kuten Eskola ja Suoranta toteavat, laadullisessa tutkimuksessa on kuitenkin hyvä muistaa, että ”aineisto kertoo ja on koottu rajatusta tapauksesta” [EsS00]. On esimerkiksi mahdollista, että valitsemalla haastateltavat vielä laajemmin erilaisista taustoista oltaisiin voitu päästä jossain määrin monipuolisempiin tuloksiin. Kuva 19 (s. 58) havainnollistaa tutkimukseen valittujen haastateltavien toimenkuvia, joissa painottuivat sovellus- ja integraatiokehittäjien roolit. Sen sijaan haastateltavissa ei ollut henkilöitä, jotka olisivat toimineet jossain vaiheessa uraansa organisaatiossa johtavissa tai yrittäjärooleissa.

Tutkimuksen suurimmaksi haasteeksi osoittautui tutkimuskohteen monitahoisuus ja käsittely. Tutkimusprosessin aikana ei löytynyt valmiita teorioita, jotka olisivat suoraan liittyneet valittuun aiheeseen eli laatua haastaviin ilmiöihin ohjelmistotuotannon päätöksenteossa. Tutkija muodosti esiyymmärryksen tutkimuskohteesta paneutumalla yleisesti laatua, IT:n hallinnointia ja päätöksentekoa käsittelevään kirjallisuuteen, jota täydennettiin ohjelmistotuotannon viitekehyksillä. Vasta haastatteluaineiston lukeminen ja analysoiminen useaan kertaan selkiyttivät, minkälaisiin teemoihin ja käsitteisiin tutkimuskohde tarkalleen ottaen kytkeytyi kirjallisuudessa. Siten tutkimuksesta kehkeytyi jossain määrin tavanomaisesta laadullisesta tutkimuksesta poikkeava, mikä näkyi teoria-aineksen jakautumisena kahteen osaan:

esiymmärrystä kuvaavaan tietoperustaan ja tuloksia täydentävään kirjallisuuteen. Tutkimuskohteen monitahoisuus ja ajalliset haasteet aiheuttivat sen, että ilmiöiden käsittely jäi pinta-puoliseksi.



Kuva 19. Havainnollistus haastateltavien toimenkuvista

Toisaalta juuri valittu näkökulma teki tutkimuksesta paitsi haastavan myös kiinnostavan. Ilmeisistä haasteista huolimatta tutkimuksessa onnistuttiin tunnistamaan ja käsitteellistämään laatua haastavia ilmiöitä, joiden ymmärtämisestä on hyötyä muun muassa työskennellessä IT:n ja liiketoiminnan rajapinnassa. Kattavan käsitteellisen mallin sijaan tutkimuksessa tehtiin perusteltua jäsenystä ilmiöistä, joista jokainen voisi itsessään toimia tutkimuskohteena. Ilmiöitä kuvaavia käsitteitä pystyttiin peilaamaan valikoituihin ohjelmistotuotannon viitekehyksiin. Myös tutkijan henkilökohtaiset tavoitteet täyttyivät: valikoidut viitekehykset tulivat huomattavasti tutummiksi ja tutkimus tuotti aidosti hyödyllistä ja käyttökelpoista materiaalia työhön.

5.3 Jatkotutkimusehdotukset

Tutkijan ehdotukset jatkotutkimusta varten ovat seuraavat: Ensinnäkin tutkimusta voitaisiin jatkaa haastattelemalla henkilöitä, joiden rooleja (kuva 19) ei ollut vielä tässä tutkimuksessa

edustettuina. Tällaisia ovat muun muassa yrityksen ylin johto, kokonaisarkkitehtuurityötä tekevät arkkitehdit sekä liiketoimintakonsultit. Haastateltavien henkilöiden valinnassa voitaisiin käyttää rajauksena myös heidän työnantajaorganisaatioidensa kokoa ja tyyppiä. Esi-merkiksi voitaisiin rajautua pk-yrityksissä tai julkishallinnossa työskenteleviin.

Toinen tunnistettu jatkotutkimusidea on tehdä syventäviä tutkimuksia löydettyihin ilmiöihin tarkastelemalla niitä omina kokonaisuuksinaan. Tässä yksi lähestymistapa voisi olla hahmottaa ilmiöitä sosioteknisinä ratkaisumalleina. Silloin kukin ilmiöistä kuvattaisiin yhdenmukaisen metamallin määrittämään rakenteeseen. Tämä toisi selkeyttä ja olisi yksi mahdollinen ratkaisutapa ilmiöiden heterogeenisen luonteen käsittelyyn.

Kolmas jatkotutkimusidea on laatia tietoperustaan valituille viitekehyksille yhdistelmäontologia. Viitekehysten ymmärtämistä ja vertailemista toisiinsa hankaloittaa se, että niiden tietosisällön rakenteisuuden aste vaihtelee paljon. Lisäksi viitekehykset eivät perustu mihinkään yhteiseen taustalla olevaan metamalliin. Narratiivista sisältöä on viitekehyksissä runsaasti, ja merkityksen esiin nostaminen siitä on työlästä. Yhdistetyllä ontologialla tavoiteltaisiin viitekehysten ymmärryksen helpottamista ja nopeuttamista. Se myös mahdollistaisi systemaattisemman tavan vertailla ja hahmottaa viitekehysten käsittemallien eroja.

Lähteet

- AKO16 Avgeriou, P., Kruchten, P., Ozkaya, I., Seaman, C. Managing Technical Debt in Software Engineering (Dagstuhl Seminar 16162). Dagstuhl Reports, 6(2016)
- Ala11 Alasuutari, P., Laadullinen tutkimus 2.0. Vastapaino, Tampere, 2011.
- AlB18 Albarak, M. ja Bahsoon R., Prioritizing technical debt in database normalization using portfolio theory and data quality metrics. Proc. The 2018 International Conference on Technical Debt, 2018, sivut 31–40.
- AnJ16 Anttila, J., Jussila, K., Mitä laatu on? https://www.sfs.fi/ajankoh-taista/uutiskirjeet/uutiskirjeet_2016/mita_laatu_on_artikkeli, 15.2.2016. [1.11.2020]
- ApL05 April, A. ja Laporte, C., An Overview of Software Quality Concepts and Management Issues. Teoksessa Measuring Information Systems Delivery Quality, Idea Group, 2005, sivut 27–52. [Myös https://www.researchgate.net/publication/236610453_An_Overview_of_Software_Quality_Concepts_and_Management_Issues, 18.10.2020].
- Ash19 Ashford, W., Zero trust: a 21st century security model. Computer-Weekly.com, 3.5.2019. <https://www.computer-weekly.com/news/252462801/Zero-trust-a-21st-century-security-model>. [31.10.2020]
- BBB01 Beck, K., Beedle, M., van Bennekum, A. et al., Ketterän ohjelmistokehityksen julistus, 2001. <https://agilemanifesto.org/iso/fi/manifesto.html>. [18.10.2020]
- Boa03 IT Governance Institute®, Board Briefing on IT Governance, 2nd edition. Rolling Meadows, IL, 2003.
- BSH15 Bartens, Y., Schulte, F., de Haes, S., Voss, S. ja Lamoën, Y., On the Way to a Minimum Baseline in IT Governance: Using Expert Views for Selective Implementation of COBIT 5. In 48th Hawaii International Conference on System Sciences, 2015.
- Cam20 Cambridge Dictionary, <https://dictionary.cambridge.org>. Hakutermit: mission-critical, business-critical [26.10.2020].
- Cap13 Caputo, A., A literature review of cognitive biases in negotiation processes. International Journal of Conflict Management, 24, 4 (2013), sivut 374–398. [Myös https://www.researchgate.net/publication/259443381_A_literature_review_of_cognitive_biases_in_negotiation_processes, 18.10.2020].

- Che20 Cherry, K., What Is Cognitive Bias? 19-July-2020.
<https://www.verywellmind.com/what-is-a-cognitive-bias-2794963>.
[18.10.2020]
- CMO72 Cohen, M. D., March, J. G., Olsen, J. P. A carbage can model of organizational choice. *Administrative Science Quarterly* 17, 1(1972), sivut 1–25.
- COB18a COBIT 2019 Framework: Introduction and Methodology, ISACA, 2018.
- COB18b COBIT 2019 Framework: Governance and Management Objectives, ISACA, 2018.
- DeM03 DeLone, W. H., McLean, E. R., The DeLone and McLean Model of Information Systems Success: a ten-year update. *Journal of Management Information Systems*, 19, 4(2003), sivut 9-30. [Myös: https://www.researchgate.net/profile/Ephraim_McLean/publication/220591866_The_DeLone_and_McLean_Model_of_Information_Systems_Success_A_Ten-Year_Update/links/5432d91c0cf20c6211bcfbe2/The-DeLone-and-McLean-Model-of-Information-Systems-Success-A-Ten-Year-Update.pdf,
27.10.2020]
- DFR11 Dickinson, H., Freeman, T., Robinson, S., Williams, I., Resource scarcity and priority-setting: from management to leadership in the rationing of health care?. *Journal of Public Money & Management*, 31, 5(2011), sivut 363-370.
- EsS00 Eskola, J. ja Suoranta, J., Johdatus laadulliseen tutkimukseen. Vastapaino, Tampere, 2000.
- FeB14 Fenton, N. ja Bieman, J., *Software Metrics: A Rigorous and Practical Approach*. Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series, CRC Press, 2014.
- FeN00 Fenton, N., Neil, M. (2000). *Software Metrics: Roadmap*. Proceedings of the Conference on the Future of Software Engineering. 10.1145/336512.336588.
- Fow09 Fowler, M., Technical Debt Quadrant 14-October-2009. <http://www.martinfowler.com/bliki/TechnicalDebtQuadrant.html>. [18.10.2020]
- Fra00 Frank, R. H., Why is cost-benefit analysis so controversial?. *The Journal of Legal Studies*, University of Chicago Press, vol. 29, 2(2000), pages 913-930.
- Fre86 Fredrickson, J. W., The strategic decision process and organizational structure. *The Academy of Management Review*, 11, 2 (1986), sivut 280–297. [Myös <https://doi.org/10.2307/258460>, 18.10.2020].
- Gar84 Garvin, D., What Does 'Product Quality' Really Mean?. *Sloan Management Review*, 15.10.1984. <https://sloanreview.mit.edu/article/what-does-product-quality-really-mean/>. [1.11.2020]

- GHW17 Günther, B., Heiligt, S., Webb, A., A case study in combating bias. Haastattelu, McKinsey Quarterly, 11.5.2017. <https://www.mckinsey.com/business-functions/organization/our-insights/a-case-study-in-combating-bias>, 2017. [23.10.2020]
- GLK14 Glazer, S., ja Karpati, T., The role of culture in decision making. Cutter IT Journal, 27, 9 (2014), sivut 23–29. [Myös https://www.researchgate.net/publication/293117922_The_role_of_culture_in_decision_making, 18.10.2020].
- Gri05 Grint, K., Problems, problems, problems: the social construction of 'leadership'. Human Relations - HUM RELAT, 58(2005), sivut 1467-1494.
- GSM04 Gallegos, F., Senft, S., Manson, D. P. ja Gonzales, G., Information Technology Control and Audit. CRC Press, 2004.
- HaC01 Halvorsen, C. ja Conradi, R. (2001). A Taxonomy to Compare SPI Frameworks. Lecture Notes in Computer Science 2077, 2001, sivut 217–235.
- HaG09 De Haes S., Van Grembergen W., Enterprise Governance of Information and Technology: Achieving Alignment and Value, featuring COBIT 5, Second Edition. Management for Professionals. Springer, Cham.
- Hak15 Hakala, J. T., Toimivan tutkimusmenetelmän löytäminen. Teoksessa Ikku-noita tutkimusmetodeihin 1, Valli, R., Aaltola, J., toimittajat, PS-kustannus, Jyväskylä, 2015, sivut 14–26.
- HiC10 Hinchey, M., Coyle, L., Evolving Critical Systems: a research agenda for computer-based systems. Proc. 17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, ECBS 2010, Oxford, England, UK, 22-26 March 2010, sivut 430-435. [Myös: <https://core.ac.uk/download/pdf/59346835.pdf>, 26.10.2020]
- HiH04 Hirsjärvi, S. ja Hurme, H., Tutkimushaastattelu: Teemahaastattelun teoria ja käytäntö. Yliopistopaino, Helsinki, 2004.
- Hin07 Hinkle, M. M., Software Quality, Metrics, Process Improvement, and CMMI: An interview with Dick Fairley. Haastattelijä: Matthew M. Hinkle. Haastateltava: Richard E. (Dick) Fairley. IT Professional, 9, 3 (2007), sivut 47-51, IEEE.
- HHS19 Hacks, S., Höfert, H., Salentin, J., Yeong, Y. C. ja Lichter, H., Towards the Definition of Enterprise Architecture Debts. 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), Paris, France, 2019, sivut 9–16.

- Hyy12 Hyöryläinen, E., Johtaminen rationaalisuuteen pyrkivänä päätöksentekona. Teoksessa: Hyöryläinen, E. (toim.), Näkökulmia hallintotieteisiin. Vaasan Yliopiston julkaisuja, sivut 42-62. Vaasa, 2012.
- HäL07 Hämäläinen, N., Liimatainen, K., A framework to support business-IT alignment in Enterprise Architecture decision making. Proc. EBRF 2007, Research Forum to Understand Business in Knowledge Society, September 25-27, Jyväskylä, Finland, 2008.
- IEE90 IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990, IEEE, 1990, sivut 1–84.
- IGI20 What is Software Process Improvement (SPI). <https://www.igi-global.com/dictionary/software-development-process-standards-for-very-small-companies/41542>. [1.11.2020]
- INS20 InsignTD Survey: Investigating causes and implications of Technical Debt. <http://www.td-survey.com/>. [18.10.2020]
- Inv20 Investopedia, <https://www.investopedia.com>. Hakutermit: mission-critical, scarcity, cost-benefit analysis, opportunity cost [26.10.2020].
- ISA20 ISACA, History of ISACA, 2020. <https://www.isaca.org/why-isaca/about-us/history>. [18.10.2020]
- ISA19 ISACA, ISACA Finland Chapter, 2019. <https://engage.isaca.org/finlandchapter/aboutchapter/about>. [18.10.2020]
- ISO11a ISO/IEC 25010:2011 Systems and Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models
- ISO11b ISO/IEC 25012:2008 Systems and Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - Data quality model
- ISO11c ISO/IEC 25012:2008 Systems and Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality requirements
- ISO15 ISO/IEC TR 19759:2015 Software Engineering — Guide to the software engineering body of knowledge (SWEBOK)
- ISO90 ISO 9000:2015 Quality management systems — Fundamentals and vocabulary
- IT417 The Open Group, The Open Group IT4IT™ Reference Architecture, Version 2.1, Open Group Standard, 2017.
- IT420 The Open Group, IT4IT Forum members list. https://reports.open-group.org/it4it_forum.shtml. [26.10.2020]

- JHS212 JUHTA - julkisen hallinnon tietohallinnon neuvottelukunta, JHS 212 ICT-palvelujen palvelutasonhallinta (SLM). Versio 1.0, 9.10.2019. <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS212/JHS212.html>. [28.10.2020]
- JoB12 Jones, C. ja Bonsignour, O., The Economics of Software Quality. Pearson Education, 2011.
- JuG98 Juran, J. M., Godfrey, A. B., Juran's Quality Handbook, 5th ed. McGraw-Hill, 1998.
- Kal02 Kaler, J., Responsibility, accountability and governance. Business Ethics: A European Review, 11, 4 (2002), sivut 327–334.
- Kan03 Kan, S., Metrics and models in software quality engineering, 2nd edition. Addison-Wesley Professional, 2003.
- Kan08 Kananen, J., Kvali: Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylän ammattikorkeakoulu, Jyväskylä, 2008.
- Kop15 Koppa, Määrällinen tutkimus 13-huhtikuuta-2015. Jyväskylän yliopisto. <https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/maarallinen-tutkimus>. [18.10.2020]
- KTS14 Khanam, S., Talib, F., Siddiqui, J., A Review of TQM and IT Research in the ICT Industry: An Agenda for Future. International Journal of Information and Computer Science (IJICS), 3 (2014). [Myös: https://www.researchgate.net/profile/Faisal_Talib/publication/267538008_A_Review_of_TQM_and_IT_Research_in_the_ICT_Industry_An_Agenda_for_Future/links/545245190cf24884d887a92e/A-Review-of-TQM-and-IT-Research-in-the-ICT-Industry-An-Agenda-for-Future.pdf, 27.10.2020]
- LaA18 Laporte, C. Y., April, A., Software quality assurance. IEEE Computer Society, Inc., Hoboken, NJ, USA, 2018
- LDM20 Lenarduzzi, V., Daly, J., Martini, A., Panichella, S. ja Tamburri, D., Towards a Technical Debt Conceptualization for Serverless Computing. IEEE Software, 2020. [Myös https://www.researchgate.net/publication/344442530_Towards_a_Technical_Debt_Conceptualization_for_Serverless_Computing, 18.10.2020].
- Lex20 Lexico, The definition of quality. <https://www.lexico.com/definition/quality>. [1.11.2020]
- LLL08 Lincke, R., Lundberg, J., Löwe, W., Comparing Software Metrics Tools. ISSTA '08: Proc. 2008 international symposium on Software testing and analysis, 2008, sivut 131-142.

- MGW18 McGrath, S. ja Whitty, J., Accountability and responsibility defined. *International Journal of Managing Projects in Business*, 11, 3 (2018), sivut 687–707. [Myös https://www.researchgate.net/publication/324582377_Accountability_and_responsibility_defined, 18.10.2020].
- MiW01 Mintzberg, H. ja Westley, F., Decision making: it's not what you think. *MIT Sloan Management Review*, 42, 3 (2001), sivut 89–93. [Myös <https://sloanreview.mit.edu/article/decision-making-its-not-what-you-think/>, 18.10.2020].
- MMZ12 McGregor, J. D., Monteith, J. Y. ja Zhang, J., Technical debt aggregation in ecosystems. 2012 Third International Workshop on Managing Technical Debt (MTD), Zurich, 2012, sivut 27–30.
- Nia06 Niazi, M., Software Process Improvement: A Road to Success. *Lecture Notes in Computer Science* 4034, 2006, sivut 395–401. [Myös https://www.researchgate.net/publication/221219024_Software_Process_Improvement_A_Road_to_Success, 18.10.2020].
- NOK12 Nord, R. L., Ozkaya, I., Kruchten, P. ja Gonzalez-Rojas, M., In search of a metric for managing architectural technical debt. 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, 2012, sivut 91–100. [Myös https://resources.sei.cmu.edu/asset_files/ConferencePaper/2012_021_001_88045.pdf, 18.10.2020].
- OMR15 Ojasalo, K., Moilanen, T. ja Ritalahti, J., Kehittämistyön menetelmät: uudenlaista osaamista liiketoimintaan. Sanoma Pro Oy, Helsinki, 2015.
- PaW96 Paloheimo, H., Wiberg, M., (1996). *Politiikan perusteet*. WSOY, Helsinki, Porvoo.
- PHA18 Pollack, J., Helm, J. ja Adler, D., What is the Iron Triangle, and how has it changed? *International Journal of Managing Projects in Business*, 11, 2 (2018), sivut 527–547.
- PMB17 Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, Sixth Edition, 2017.
- PMI20 Learn about PMI, Project Management Institute. <https://www.pmi.org/about/learn-about-pmi>. [24.10.2020].
- Por85 Porter, M. E. *Competitive Advantage: Creating and Sustaining Superior Performance*. New York: Simon and Schuster, 1985.
- RMS18 Rios, N., Mendonça, M. ja Spínola, R., A Tertiary Study on Technical Debt: Types, Management Strategies, Research Trends, and Base Information for

- Practitioners. Information and Software Technology, 102 (2018), sivut 117–145.
- Sap06 Saaranen-Kauppinen, A. ja Puusniekka, A., Teemahaastattelu. KvaliMOTV – Menetelmäopetuksen tietovaranto, Tampere, 2006.
https://www.fsd.tuni.fi/menetelmaopetus/kvali/L6_3_2.html. [18.10.2020]
- Sim79 Simon, H. A., Rational decision making in business organizations. The American Economic Review, 69, 4(1979), sivut 493-513.
- Sli08 Sliger, M. (2008). Agile project management and the PMBOK® guide. Paper presented at PMI® Global Congress 2008—North America, Denver, CO. Newtown Square, PA: Project Management Institute.
- Sta76 Staw, B. M., Knee-deep in the Big Muddy: A Study of Escalating Commitment to a Chosen Course of Action. Organizational Behavior and Human Performance, 16, 1(1976), sivut 27–44.
- Ste95 Stevens, T., Philip Crosby: Quality is still free. Haastattelu. Haastattelija: Tim Stevens. Haastateltava: Philip Crosby. IndustryWeek, 19.6.1995. [Myös: <https://www.industryweek.com/operations/quality/article/21964139/philip-crosby-quality-is-still-free>, 29.10.2020]
- SWE14 SWEBOK v3.0: Guide to the software engineering body of knowledge. IEEE, 2014. <https://www.swebok.org>. [25.10.2020]
- SWE20 SWEBOK Evolution. <https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/software-engineering-committee/swebok-evolution>, 2020. [25.10.2020]
- TaF18 Tambo, T., Filtenborg, J., Digital services governance: IT4IT™ for management of technology. Journal of Manufacturing Technology Management, 10, 2018. [Myös: <https://www.emerald.com/insight/content/doi/10.1108/JMTM-01-2018-0028/full/html>, 26.10.2020]
- Tia05 Tian, J., Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement. IEEE Computer Society, 2005. Hoboken, NJ, USA.
- TKL15 Tamburri, D. A., Kruchten, P., Lago, P., van Vliet, H., Social debt in software engineering: insights from industry. Journal of Internet Services and Applications, 6, 10 (2015)
- TOG18 The Open Group, The TOGAF® Standard, Version 9.2, 2018.
- TOG19 The Open Group, Enabling a smarter DevOps culture with the IT4IT standard. The Open Group Blog. <https://blog.opengroup.org/2019/12/03/enabling-a-smarter-devops-culture-with-the-it4it-standard/> [26.10.2020]

- TOG20 The Open Group, TOGAF® Library. <https://publications.opengroup.org/to-gaf-library>, 2020. [25.10.2020]
- TuS03 Tuomi, J. ja Sarajärvi, A., Laadullinen tutkimus ja sisällönanalyysi. Tammi, Helsinki, 2003.
- Tör02 Törrönen, J., Semiotic theory on qualitative interviewing with stimulus texts. *Qualitative Research*, 2, 3 (2002), sivut 343–362. [Myös https://www.researchgate.net/publication/236480247_Semiotic_theory_on_qualitative_interviewing_with_stimulus_texts, 18.10.2020].
- UDL08 Umoquit, M. J., Dobrow, M. J., Lemieux-Charles, L., Ritvo, P. G., Urbach, D. R. ja Wodchis, W. P., The efficiency and effectiveness of utilizing diagrams in interviews: an assessment of participatory diagramming and graphic elicitation. *BMC Medical Research Methodology*, 8, 53 (2008). [Myös https://www.researchgate.net/publication/23160108_The_efficiency_and_effectiveness_of_utilizing_diagrams_in_interviews_An_assessment_of_participatory_diagramming_and_graphic_elicitation, 18.10.2020].
- UnB09 Uncle Bob, A Mess is not a Technical Debt 22-September-2009. <https://sites.google.com/site/unclebobconsultingllc/a-mess-is-not-a-technical-debt>. [18.10.2020]
- UTB11 Umoquit, M. J., Tso, P., Burchett, H. ja Dobrow, M. J., A multidisciplinary systematic review of the use of diagrams as a means of collecting data from research subjects: application, benefits and recommendations. *BMC medical research methodology*, 11, 11 (2011). [Myös <https://bmcmmedresmethodol.biomedcentral.com/articles/10.1186/1471-2288-11-11>, 18.10.2020].
- Vil15 Vilkkä, H., Tutki ja kehitä. PS-kustannus, Jyväskylä, 2015.
- Wei95 Weick, K. E., Sensemaking in Organizations. SAGE Publications, 1995.
- Wie96 Wiegers, K. E., Creating a Software Engineering Culture. Dorset House Publishing, New York, 1996.
- WeR04 Weill, P. ja Ross, J. W., IT Governance: How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, Boston, 2004.
- ZWI13 Zazworka, N., Vetro, A., Izurieta, C., Wong, S., Cai, Y., Seaman, C. ja Shull, F., Comparing four approaches for technical debt identification. *Software Quality Journal*, 22, 3 (2013), sivut 1–24. [Myös https://www.researchgate.net/publication/256708004_Comparing_Four_Approaches_for_Technical_Debt_Identification, 18.10.2020].